PB80-178361

Evaluation of the Real-Time Air-Quality Model Using the RAPS (Regional Air Pollution Study) Data Base. Volume 3. Program User's Guide

SRI International
Menlo Park, CA

Prepared for

Environmental Sciences Research Lab.
Research Triangle Park, NC

Feb 80

U.S. Department of Commerce
National Technical Information Service
NTIS

United States
Environmental Protection
Agency

Environmental Sciences Research
Laboratory
Research Triangle Park NC 27711

EPA-600/4-80-013c
February 1980

PB80-178361

Research and Development

EPA

Evalution of the Real-Time Air-Quality Model Using the RAPS Data Base

Volume 3.
Program User's Guide

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

RESEARCH REPORTING SERIES

Research reports of the Office of Research and Development, U.S. Environmental Protection Agency, have been grouped into nine series. These nine broad categories were established to facilitate further development and application of environmental technology. Elimination of traditional grouping was consciously planned to foster technology transfer and a maximum interface in related fields. The nine series are:

1. Environmental Health Effects Research
2. Environmental Protection Technology
3. Ecological Research
4. Environmental Monitoring
5. Socioeconomic Environmental Studies
6. Scientific and Technical Assessment Reports (STAR)
7. Interagency Energy-Environment Research and Development
8. "Special" Reports
9. Miscellaneous Reports

This report has been assigned to the ENVIRONMENTAL MONITORING series. This series describes research conducted to develop new or improved methods and instrumentation for the identification and quantification of environmental pollutants at the lowest conceivably significant concentrations. It also includes studies to determine the ambient concentrations of pollutants in the environment and/or the variance of pollutants as a function of time or meteorological factors.

This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

TECHNICAL REPORT DATA

(Please read Instructions on the reverse before completing)

1. REPORT NO. EPA-600/4-80-013c	2.	3. RECIPIENT'S ACCESSION NO. PB 80 178361
4. TITLE AND SUBTITLE EVALUATION OF THE REAL-TIME AIR-QUALITY MODEL USING THE RAPS DATA BASE Volume 3. Program User's Guide		5. REPORT DATE February 1980 6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) R.E. Ruff, H. Shigeishi, and R.H. Allen (Comp-Aid, Inc.)		8. PERFORMING ORGANIZATION REPORT NO. Final Report SRI Project 6868
9. PERFORMING ORGANIZATION NAME AND ADDRESS SRI International 333 Ravenswood Avenue Menlo Park, California 94025		10. PROGRAM ELEMENT NO. 1AA603 AA-26 (FY-77) 11. CONTRACT/GRANT NO. 68-02-2770
12. SPONSORING AGENCY NAME AND ADDRESS Environmental Sciences Research Laboratory — RTP, NC Office of Research and Development U.S. Environmental Protection Agency Research Triangle Park, North Carolina 27711		13. TYPE OF REPORT AND PERIOD COVERED FINAL 8/77-4/79 14. SPONSORING AGENCY CODE EPA/600/09

15. SUPPLEMENTARY NOTES

16. ABSTRACT

The theory and programming of statistical tests for evaluating the Real-Time Air-Quality Model (RAM) using the Regional Air Pollution Study (RAPS) data base are fully documented in four volumes. Moreover, the tests are generally applicable to other model evaluation problems. Volume 3 presents the software used in the statistical tests for evaluating the RAM. Six statistical tests are described, with attention to the programming philosophy behind them. Also presented is a review of the auxiliary software that sort, retrieve, format, and display the data.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group
* Air pollution * Mathematical models * Evaluation * Tests * Computer systems programs * Statistical tests	Real-Time Air-Quality Model Regional Air Pollution Study Data Base	13B 12A 14B 09B

18. DISTRIBUTION STATEMENT RELEASE TO PUBLIC	19. SECURITY CLASS *(This Report)* UNCLASSIFIED	21. NO. OF PAGES
	20. SECURITY CLASS *(This page)* UNCLASSIFIED	22. PRICE

EPA-600/4-80-013c
February 1980

EVALUATION OF THE REAL-TIME AIR-QUALITY MODEL USING THE RAPS DATA BASE

Volume 3. Program User's Guide

by

Ronald E. Ruff and Hisao Shigeishi
Atmospheric Science Center
SRI International
Menlo Park, California 94025

and

Rodney H. Allen
Comp-Aid, Inc.
Research Triangle Park, North Carolina 27711

Contract No. 68-02-2770

Project Officer

John S. Irwin
Meteorology and Assessment Division
Environmental Sciences Research Laboratory
Research Triangle Park, North Carolina 27711

ENVIRONMENTAL SCIENCES RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27711

1a

DISCLAIMER

This report has been reviewed by the Environmental Sciences Research Laboratory, U.S. Environmental Protection Agency, and approved for publication. Approval does not signify that the contents necessarily reflect the views and policies of the U.S. Environmental Protection Agency, nor does mention of trade names or commercial products constitute endorsement or recommendation for use.

ABSTRACT

The theory and programming of statistical tests for evaluating the Real-Time Air-Quality Model (RAM) using the Regional Air Pollution Study (RAPS) data base are fully documented in four volumes. Moreover, the tests are generally applicable to other model evaluation problems. Volume 3 presents the software used in the statistical tests for evaluating the RAM. Six statistical tests are described, with attention to the programming philosophy behind them. Also presented is a review of the auxiliary software that sort, retrieve, format, and display the data. This report was submitted in fulfillment of Contract No. 68-02-2770 by SRI International under the sponsorship of the U.S. Environmental Protection Agency. This report covers a period from 1 October 1977 to 1 April 1979, and work was completed as of 1 April 1979.

CONTENTS

Abstract iii
Figures vi
Tables viii

1. Overview 1
    Background 1
    Discussion of statistical programs 2
    Programming philosophy 3
2. Data-Base Considerations 4
    Content 4
    Structure and formats 4
    Display and retrieval software 10
    Auxiliary software 13
3. Statistical Package. 16
    Introduction 16
    Accuracy score 16
    Residual time series 30
    Chi-square goodness-of-fit 34
    Bivariate regression and correlation 38
    Interstation error correlation 43
    Multiple regression of error residuals 45

References 50
Appendices

A. Listing of test data base 51
B. Modified EPA frequency-distribution software 54
C. Accuracy score listings 66
D. Residual time series listings 93
E. Chi-square goodness-of-fit listings 105
F. Bivariate regression and correlation listing 114
G. Interstation error correlation listings 125

Preceding page blank

FIGURES

Number **Page**

1 SPSSDAT program 8

2 SPSSSORT program 11

3 Examples of SPSS histograms 12

4 Frequency distribution of observed and predicted concentrations on log-probability axes 14

5 Frequency distributions of residuals (percent) on log-probability axes 15

6 Accuracy score equations 17

7 Flow diagram for the accuracy-score program 19

8 Results of Test 1 displayed on a map of St. Louis 21

9 First part of the accuracy score session 22

10 Accuracy score session (MODE CODE = 1) 23

11 Accuracy score session (MODE CODE = 2) 24

12 Software structure for the accuracy score program 25

13 Display of the autocorrelation function for Station 1 of the test data base 31

14 Display of the normalized cumulation periodogram for Station 1 of the test data base 32

15 Program structure for the residual time series 33

16 Histogram of concentrations for the test data base 35

17 Histogram of concentration residuals for the test data base 36

18 Structure of the chi-square goodness-of-fit program 37

Number	Page

19 Scattergram with least-squares, confidence, and probability lines 39

20 Structure of the regression and correlation program 40

21 COREL2 output for test data base 43

22 Structure of the interstation error correlation program 44

23 Example of a regression procedure 46

24 Summary output for regression of RESID on ASC, WDC, and WSC 47

25 Summary output for regression of RESID on TIME and SITE 47

26 Example of SPSS SCATTERGRAM output 49

TABLES

Number Page

1 Definition of User-Entered Values for the Accuracy Score Test 23

2 Description and Format of Test Data Base 27

SECTION 1

OVERVIEW

BACKGROUND

The programs described in this volume are meant to be general-purpose tools useful to anyone with a computer supporting the FORTRAN IV programming language and a graphics display. However, from a practical standpoint, we had to tune our programming efforts to EPA's National Computer Center (NCC) in Research Triangle Park, North Carolina. The mainframe is a Univac 1110 computer. (The executive system, Exec 8, is described in Volume VII of the NCC User Reference Manual.[1]) Standard peripherals were used whenever possible. For graphics, the Tektronix 4014 terminal is the only device on which the plotting programs can operate. (Graphic output in this text are from the 4014 using a Tektronix 4631 hardcopy unit.)

Most of the programs were written in Univac's FORTRAN V language. (See Volume III of the NCC User Reference Manual.) Certainly the programs as they now exist would require only minimal effort for conversion to FORTRAN IV. For graphics, FORTRAN-callable Tektronix subroutines in the Advanced Graphing II[2] and the Terminal Control System[3] manuals were used. A few programs were written in the special Statistical Packages for the Social Sciences (SPSS) language.[4] Many major computer facilities have this software package or one that is quite similar.

From the above description, it should be clear that the software presented will operate only on the NCC system. However, with minimal modification it should be readily adaptable to most other major computer systems. For the remainder of this text, we assume that the reader is familiar with the hardware and basic software used. We do attempt to point out code that is usually specific to the NCC computer system.

DISCUSSION OF STATISTICAL PROGRAMS

The six statistical tests recommended in Volume 2 were programmed, debugged, and tested. The tests and program names are given below:

Test Name	Program Name[*]
Accuracy score	M21AD0*STAT01.ACSCOR
Residual time series	M21AD0*STAT02.TIMSER
Chi-square goodness-of-fit	M21AD0*STAT03.CHIFIT
Bivariate regression and correlation[†]	M21AD0*STAT04.REGANA
Interstation error Correlation	M21AD0*STAT05.COREL2
Multiple regression of error residuals	M21AD0*STAT06.SPSSRUN

The Pearson's correlation coefficient and linear regression tests were combined because (1) it was computationally efficient to do so and (2) results from both tests have historically been presented together when evaluating air-quality models. The six programs are fully described in Section 3 of this volume. (Note that the 0's are zeros in M21AD0 and STAT01 through STAT06 and are the letter O elsewhere.)

---

[*] The program names are given in typical Exec 8 file designation format--FILENAME.ELEMENT. On the NCC computer, one must often qualify the file with an account code--ACCOUNT*FILENAME.ELEMENT. The account qualifier for these statistical programs is M21AD0. All the source, relocatable, and absolute elements found in files M21AD0*STAT01 through M21AD0*STAT06 are backed up in a single file labeled SRI*SRI. In this file an absolute such as M21AD0*STAT01.ACSCOR is also known as SRI*SRI.ACSCOR. The test data base in data element M21AD0*STAT01.COQS2N is also known as SRI*SRI.COQS2N. All other source, data, and relocatable elements follow the example for M21AD0*STAT02.CASE2 which is also contained in SRI*SRI.CASE2/STAT02.

[†] Combines the features of two tests: (1) linear regression with confidence and prediction bands and (2) Pearson's correlation coefficient. See Volume 2 for a complete description.

Each of the program files (STAT01 through STAT06)* contains all the subroutines and run streams (control language statements) required to run the statistical test in question. In addition, a test data base is contained in M21ADO*STAT01.COQS2N. All programs except those contained in STAT06 are written in FORTRAN V. STAT06 programs are written in the SPSS language. In addition to the multiple regression program, SPSS is also used to plot frequency distributions of any variable, to plot scattergrams, and to retrieve subsets of the data base. These auxiliary data-retrieval and display programs are discussed in Section 2.

PROGRAMMING PHILOSOPHY

Before discussing the details of the software, we wish to impart some of the philosophy used during development. Our approach is a compromise between practical considerations in computing efficiency and the ideal software from the users' perspective.

The ideal programs would never need to be modified regardless of the data base used and the application in question. Furthermore, they would work on a large number of computers, given that these computers were equipped with certain software.

Having defined our concept of programs ideal to a user community, we now alert the reader that our programs do not satisfy that definition. However, we have standardized where practical, and through the documentation provided in this volume, the user can readily customize the programs to his application and system, assuming that the system supports FORTRAN IV, Tektronix graphics, and SPSS. (As stated earlier, we attempt to point out sections of the program that are specific to the NCC Univac system.)

---

*For brevity, the account qualifier--M21ADO--is often omitted.

SECTION 2

DATA-BASE CONSIDERATIONS

CONTENT

In their simplest form, the data for most statistical tests consist of n pairs of observed and predicted concentrations. These data are denoted $(OC, PC),(OC_2,PC_2),\ldots,(OC_n,PC_n)$. (This data set may include all the data gathered or only a selected subset thereof.) The concentrations might refer to hourly averages, three-hour averages, daily averages, etc., at a particular monitoring station or from any number of stations. Except for the residual time series test, the data pairs need not be sequential nor time ordered in any particular fashion.

Since it is desirable to assess model performance under a variety of meteorological and emission conditions, the data base should be arranged so that it can be subdivided into many nonexclusive subsets, according to the following parameters:

- Date, D
- Time period, T
- Location, L
- Pasquill-Gifford stability class, G_1
- Wind-speed class, G_2
- Wind-direction class, G_3
- Mixing-height class, G_4
- Emission class, Q
- Observed concentration, OC
- Predicted concentration, PC.

STRUCTURE AND FORMATS

General Structure

The exact structure (or hierarchy) of the data depends on the features of the data base to be tested. Consequently, the format chosen for our statistical programs was determined by features of our test data base. The data file (in card image form) is ordered as follows:

Card/Line Number	Data
1	$(D,T,L,G_1,G_2,G_3,G_4,Q,OC,PC)$
2	$(D,T,L,G_1,G_2,G_3,G_4,Q,OC,PC)_2$
⋮	⋮
n	$(D,T,L,G_1,G_2,G_3,G_4,Q,OC,PC)_n$

(We will refer to each record by line number rather than card number, since normally the records will be stored on disk.)

The only requirement for this structure is that all the observed and predicted concentrations--(OC,PC) through (OC_n,PC_n)--be distinct. Now, we must recognize that the data file can include concentrations for a number of monitoring locations, denoted by L. Also, for each time period, T, we may choose to concern ourselves with only one set of meteorological and emission parameters. (This would typically be the case when evaluating the RAM.) Therefore, if we expand our subscript notation somewhat, our data file would typically be ordered as follows:

Card/Line Number	Data
1	$D_1,T_1,1,(G_1,G_2,G_3,G_4,Q)_1,OC_1,PC_1$
2	$D_1,T_1,2,(G_1,G_2,G_3,G_4,Q)_1,OC_2,PC_2$
⋮	⋮
ns	$D_1,T_1,ns,(G_1,G_2,G_3,G_4,Q)_1,OC_{ns},PC_{ns}$
ns + 1	$D_2,T_2,1,(G_1,G_2,G_3,G_4,Q)_2,OC_{ns+1},PC_{ns+1}$
⋮	⋮
2*ns	$D_2,T_2,ns,(G_1,G_2,G_3,G_4,Q)_2,OC_{2*ns},PC_{2*ns}$
⋮	⋮
p*ns	$D_p,T_p,ns,(G_1,G_2,G_3,G_4,Q)_p,OC_{p*ns},PC_{p*ns}$

where ns = number of monitoring stations (locations), p = number of time periods, and ns*p = n = number of observed and predicted concentration pairs. Generally speaking, our test data base is structured as above.

While the structure described above is functional, it is probably not always computationally efficient. This is particularly true if we must apply the programs to a large data base (such as that for the RAPS). The inefficiency stems from having to store the time, date, meteorological, and emissions data once for each monitoring site. Thus repetition consumes a moderate amount of storage space. (In a one-year period, we are storing over 50,000 words of redundant data.) Consequently, for larger data bases, we would recommend storing the data in a more efficient manner.

For the RAPS data base, we note that meteorological data are collected at each of the monitoring stations. As discussed in Volume 4, it may be desirable to store these station-specific meteorological data for use in our analysis even though these data are not used directly by the model. Therefore, depending on the application, we may wish to retain additional data that will require format changes. In Volume 4, we discuss specific recommendations for evaluating the RAM using the RAPS data base.

As suggested in the above discussion, the exact structure (and even content) of the data base may vary according to application, user convenience, and computational efficiency. It generally requires modification of only one or two lines in the program code to accommodate a slightly different data structure. When we discuss the individual programs (in Section 3), we point out the sections of code that control the database interface and the exact data requirements. We stress that the user should have a flexible attitude concerning the exact format, allowing for changes that befit the application at hand.

Test Data Base

Standard (FORTRAN) Format--

The data base is stored in free-form format in STAT01.COQS2N (file.element). Appendix A contains a listing of the data. In all, there are 101 line (or card) images. Each line image contains the following information, in the order presented:

- Date--six digits
- Time--four digits
- Station--one digit whose value is 1, 2, or 3
- Atmosphere stability--one digit whose value varies from 1 to 6
- Wind-speed class--one digit whose value varies from 1 to 6
- Wind-direction class--two digits whose value varies from 1 to 16
- Observed concentration--a number between 0.0 and 1000.0 (one significant figure to the right of the decimal point)
- Predicted concentration--same format as above.

The mixing-height class and emissions class were not required when the test data base was acquired. (Refer to Volume 1 for a description of the field study during which the test data base was acquired.)

SPSS Format--

The SPSS system of computer programs is used to perform the multiple linear regression test, which relates meteorological parameters to observed and predicted concentration residuals. The SPSS is also useful in retrieving and displaying subsets of the data base. (This is described on pages 10 and 11.) Prior to using any of the SPSS procedures, the input data must be converted to a special SPSS format. This can be done within each program or, alternatively, done once and then stored in the special SPSS format. This latter approach is more efficient when an assortment or SPSS programs is continually applied. In the next paragraph, we discuss the program that converts a standard FORTRAN file into an SPSS file.

The program to reformat the data is stored in STAT06.SPSSDAT. (In discussing the SPSS procedures and codes, the reader should refer to the SPSS manual[4] for a complete description. Our descriptions will be brief and will not cover all the features of the statements involved.) The program stored in SPSSDAT appears in Figure 1 below.

```
@DELETE HSPRNT.
@ASG,PU HSPRNT.
@ASG,A SPSSINPUT.
@USE 8.,SPSSINPUT.
@ASG,PU SPSSFILE.
@USE 4.,SPSSFILE.
@BRKPT PRINT$/HSPRNT
@SPSS
RUN NAME         TEST MODEL
FILE NAME        SPSSDATA
VARIABLE LIST    DATE,TIME,SITE,ASC,WSC,WDC,OC,PC
INPUT MEDIUM     TAPE
N OF CASES       UNKNOWN
INPUT FORMAT     FIXED    (12X,2F6.0,F4.0,2F3.0,F4.0,7X,2F8.1)
MISSING VALUES   ALL (BLANKS)
COMPUTE          RESID = OC-PC
VAR LABELS       RESID,RESIDUAL
                 ASC, STABILITY CLASS
                 WSC, WIND SPEED CLASS
                 WDC, WIND DIRECTION CLASS
                 OC, OBSERVED CONCENTRATION
                 PC, PREDICTED CONCENTRATION
LIST CASES       CASES = 1000/
                 VARIABLES = ALL
READ INPUT DATA
SAVE FILE
FINISH
@BRKPT PRINT$
@FREE HSPRNT.
@SYM,U HSPRNT.,,FD04PR
```

Figure 1. SPSSDAT program.

Note that two files were assigned and related to peripheral devices (4 and 8). The first file SPSSINPUT is simply a copy of the test data base contained in STAT01.COQS2N. (SPSS cannot work directly with data contained in elements. Therefore, a system utility was used to copy data from STAT01.COQS2N to SPSSINPUT.) The second file, SPSSFILE, is the output file, which will become the specially formatted SPSS file used in subsequent SPSS programs. The @SPSS control card is the manner in which the NCC system calls the complete SPSS package. Subsequent cards will then be recognized as SPSS procedure cards. The remainder of the Exec 8 statement* control the routing of the print file, HSPRNT, to the desired terminal (FD04PR in this case).

In Figure 1, the statements following @SPSS are the SPSS control statements. (These begin in column 1. The specification of the desired parameters begins in column 16.) In our example, the RUN NAME is arbitrarily titled TEST MODEL, which thus becomes an identifier/header on the output listing. The FILE NAME has been called SPSSDATA, which becomes the output SPSS file referred to by future SPSS programs. Through the VARIABLE LIST statement we choose the desired parameter names to correspond with the data in our input file. The INPUT MEDIUM, for Exec 8, recognizes only a CARD or TAPE specification. TAPE simply identifies that the input will be read from a mass storage file, always named "8". (Thus, through the @USE statement, SPSSINPUT is identified to SPSS as the input file.) The N OF CASE statement that follows can be set to UNKNOWN if cards are not used for the input. By specifying FIXED on the INPUT FORMAT statement, we are able to use a standard FORTRAN format for the parameters in the VARIABLE LIST statement.

The preceding paragraph fully specifies the input file. Next, we describe how SPSS is used to manipulate the data and store them on the output file. First, by the MISSING VALUES statement, we could assign values to any parameters when they are missing in the input data. In our case, this is not a consideration, so we leave such data blank. The COMPUTE statement enables us to compute any desired parameters. Since we use concentration residuals in another SPSS program, we compute the RESID variable as shown. •

---

*Alternatively, for the NCC only, routing of the output file can also be controlled by using the @AB*US.SUSPEND and @AB*US.RESUME statements as shown in Figure 2, page 11.

The VAR LABELS statement is used to assign meaningful names to the parameters and subsequently cause these parameters to be better documented in the ouput of the program. The LIST CASES enables us to produce an output listing of a user-specified number of cases (e.g., 1000) and the desired variables (e.g., ALL).

The READ INPUT DATA statement instructs the program to begin reading the data. The SAVE FILE instructs the system to save the processed file. Using Exec 8, SPSS expects that the file available for writing will be named "4". (Thus, the @USE statement causes the file to be called SPSSFILE.)

The above SPSS program is by no means the only way to generate an SPSS file. Some of the statements are specified in a somewhat arbitrary fashion. We leave it to the reader to further familiarize himself with the SPSS statements involved.

DISPLAY AND RETRIEVAL SOFTWARE

Prior to selecting, applying, or interpreting a particular test, we may wish to examine the data base in some detail. For instance, we may wish to stratify the data for a given meteorological class and then run the test on that subset; or we may wish to closely examine the meteorological conditions associated with concentration comparisons that fall outside the 90 percent probability bounds in a regression analysis.

A number of computer languages could be used to satisfy the above types of requirements. Standard FORTRAN programs could be written or a data-base management language (e.g., System 2000[5]) could be applied. For our immediate needs, the most attractive choices of procedures are those available within SPSS. Two general SPSS data-retrieval and display programs are discussed in the following paragraphs.

A program stored in STAT06.SPSSSORT is an example of how SPSS can be used to selectively sort, retrieve, and display data in an SPSS file. The program is given in Figure 2 below.

```
@ASG,A HGSSORT.
@USE 15.,HGSSORT.
@AB*US.SUSPEND
@ASG,A SPSSFILE.
@USE 3.,SPSSFILE.
@SPSS
RUN NAME          SORT FILE
GET FILE          SPSSDATA
RAW OUTPUT UNIT 15
*SELECT IF        (ASC EQ 4)
WRITE CASES       (2F6.0,F4.0,2F3.0,F4.0,7X,2F8.1) DATE,TIME,SITE,ASC
                  WSC,WDC,OC,PC
*SELECT IF        (ASC EQ 4 AND SITE EQ 1)
FREQUENCIES       INTEGER = WDC(1,16) WSC(1,6)
OPTIONS           3,8
FINISH
@AB*US.RESUME,P FE04PR
```

Figure 2. SPSSSORT program.

The input file, SPSSFILE, is named "3". (Note the @USE control statement.) Within the SPSS program, the GET FILE statement automatically looks to Unit 3 for the input. SPSSDATA is the file name previously assigned by the SPSSDAT program (see Figure 1). The RAW OUTPUT UNIT 15 statement, coupled with the @ASG and @USE Exec 8 statements, identifies HGSSORT as the output file. (Note that the SPSS output will omit decimal points. Hence, the output as written must be divided by 10 in the FORTRAN program. Alternatively, we could drop the least significant figure and use integers.) The first *SELECT IF statement, followed by WRITE CASES, causes an output file to be written on HGSSORT in the format and order specified in the WRITE CASES field. The resulting file will contain those cases where the atmospheric stability classes (ASC) were class 4. This file can then be read by a FORTRAN program.

The second *SELECT IF statement, followed by the FREQUENCIES statement, causes the SPSS to generate a histogram on the line printer for the identified variables--wind direction and speed--for station 1 and an atmospheric stability class of 4. The resulting output is shown in Figure 3. OPTION 3 causes the output to be left-justified on an 8-1/2 by 11 format; OPTION 8 causes a histogram to be printed for each variable specified on the FREQUENCIES statement.

```
06 OCT 78          FILE - SPSSDATA - CREATED 04 OCT 78          PAGE   2

WDC
   CODE
        I
     1  ************ (     5)
        I
        I
        I
     2  ************************************************** (    20)
        I
        I
        I
     3  ****** (     2)
        I
        I
        I.........I.........I.........I.........I.........I
        0         4         3         12        16        20
        FREQUENCY

VALID CASES      27      MISSING CASES      0

05 OCT 78          FILE - SPSSDATA - CREATED 04 OCT 78          PAGE   4

WSC
   CODE
        I
     1  ********** (     4)
        I
        I
        I
     2  ************************* (    10)
        I
        I
        I
     3  ******************************** (    13)
        I
        I
        I.........I.........I.........I.........I.........I
        0         4         8         12        16        20
        FREQUENCY

VALID CASES      27      MISSING CASES      0
```

Figure 3. Examples of SPSS histograms.

There are a number of variations that can be used with the above program. Some of these are considered in Volume IV, in which we describe techniques for applying the statistical tests.

AUXILIARY SOFTWARE

As will be discussed thoroughly in Volume 4, simple graphic comparisons can provide very useful information during the model-evaluation process. Examples, plotted on a Tektronix 4014 terminal, are shown in Figures 4 and 5. The first graph (Figure 4) displays on logarithm-probability axes the frequency distribution of both observed and predicted concentrations; the second graph (Figure 5) displays the frequency distribution of the percentage difference (residual) in observed and predicted concentrations.

The programs to draw the axes and plot the data were developed by EPA. SRI's subcontractor, Comp-Aid Incorporated, adapted the program to our test data base and added some annotations. Options for three different graphs can be exercised. (An example of the third type of graph, frequency distribution of the absolute value of the concentration residual, is discussed in Volume 4.) The listings of the programs are given in Appendix B. The mapped (or linked) program is stored in SRI*SRI.FREQ (account*file.element). The run stream, stored in SRI*SRI.DATAFREQ, which shows its execution with our test data base (contained in M21ADO*STAT01.COQS2N) is given in Section 1 of Appendix B.

The figure for SRI*SRI.MAPFREQ, Section 2 of Appendix B, shows how the program is "mapped" and serves to identify the subroutines. The main program, SRI*SRI.FREQ, is given the name of the executable. Its listing is given in Section 3 of Appendix B. The subroutine listings are given in Section 4 of Appendix B. The principal subroutines are contained in SRI*SRI.ORDER, SRI*SRI.GRAPH, and SRI*SRI.GRALIN. The other subroutines, SRI*SRI.BOX, SRI*SRI.HEADER, and RAPS*UTILITY.COMPOZ, are used to draw the frame around the plot and annotate the data executed (upper-right-hand corner in Figures 4 and 5). These last three subroutines are discussed further in Section 3.

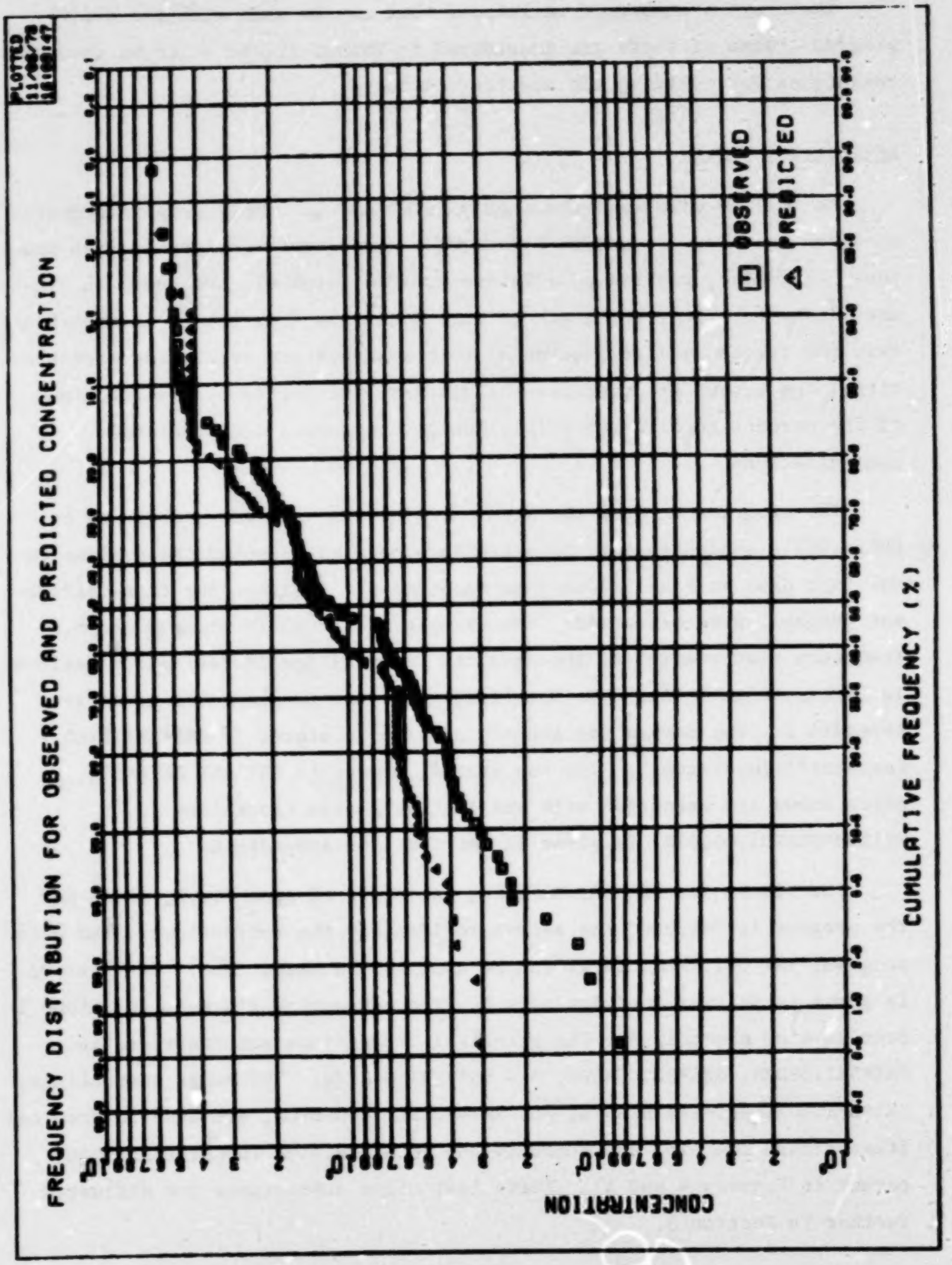


Figure 4. Frequency distribution of observed and predicted concentrations on log-probability axes.

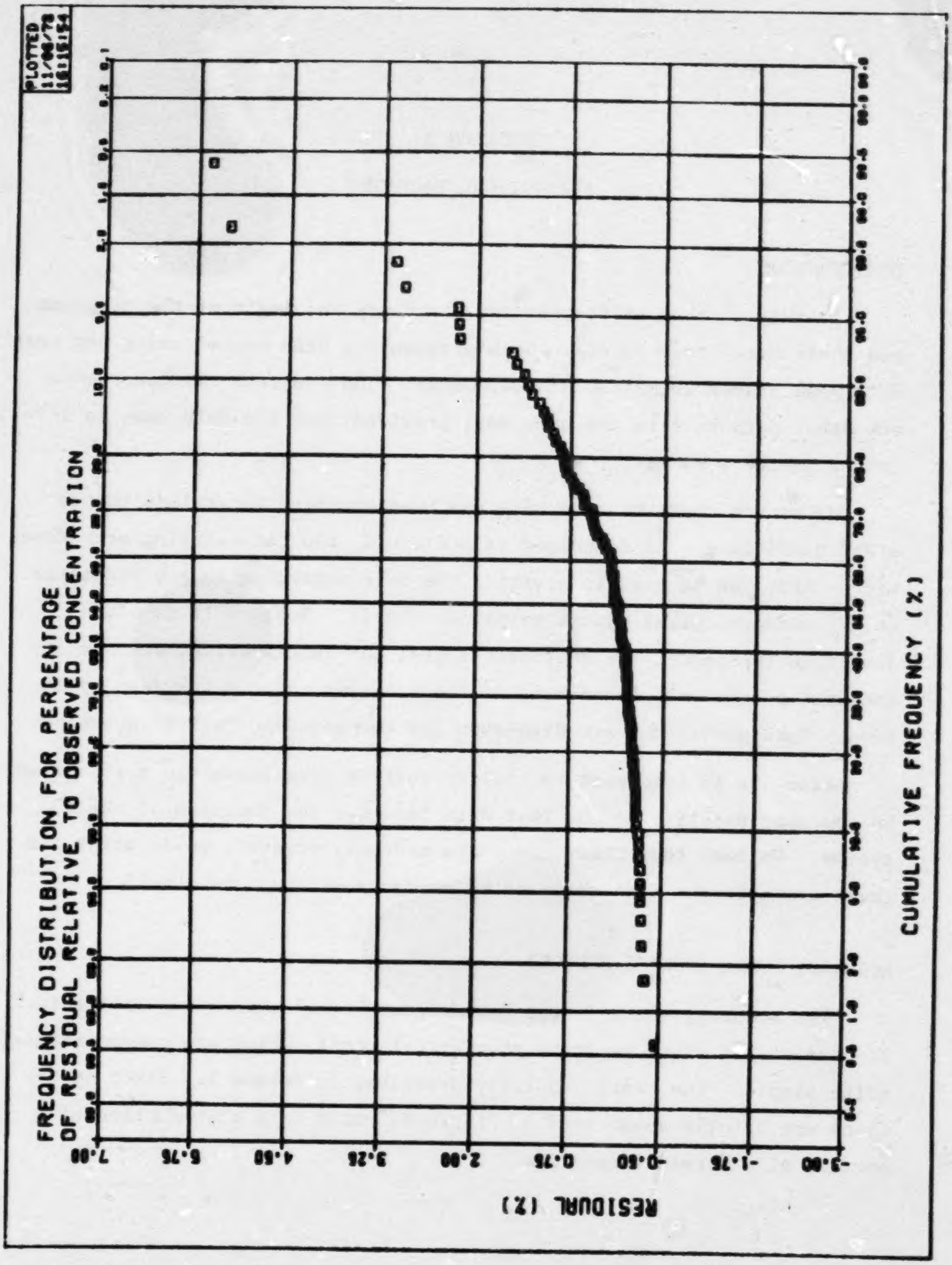


Figure 5. Frequency distributions of residuals (percent) on log-probability axes.

SECTION 3

STATISTICAL PACKAGE

INTRODUCTION

In this section we discuss the structure and logic of the programs and their data requirements. Each program has been tested using our test data base stored in M21ADO*STAT01.COQS2N. The programs can operate on any other data base in the same way, provided that the data base is formatted in the same way.

We assume that the data base has been arranged to include the desired conditions. As discussed in Section 2, the data-sorting procedures within SPSS can be used to stratify the data according to any parameter (e.g., meteorological class, emissions class). Because it is often desirable to conduct the statistical test for each station, we have included provisions within some of the programs to sort the data by station. Such provisions are discussed for the specific test in question.

Also, it is important to realize that in some cases the software may be somewhat specific to the test data base and the features of the NCC system. We feel that these cases are minimal; however, we do point out these areas in our discussion or by comments in the code itself.

ACCURACY SCORE (STAT01.ACSCOR)

The accuracy score is the recommended "final evaluation statistic." It consists of eight separate statistical tests, which are computationally quite simple. The tests are fully described in Volume 2. Their equations are briefly summarized in Figure 6, which is a reproduction of a portion of a terminal session.

```
* SELECT ONE TEST NO. FROM THE FOLLOWING ...

     1.   MEAN ABSOLUTE ERROR ...
          E = (1.0/N)*(SUM(ABS(OC(I)-PC(I))))          WHERE I=1,N
     2.   MEAN SQUARE ERROR ...
          E = (1.0/N)*(SUM((OC(I)/PC(I))**2))          WHERE I=1,N
     3.   ABSOLUTE ERROR THRESHOLD ...
          E = (1.0/N)*(SUM(C(I)))                      WHERE I=1,N
               WHERE C(I) = 0         IF ABS(OC(I)-PC(I)).LE.E1
               WHERE C(I) = 1         IF ABS(OC(I)-PC(I)).GT.E1
     4.   PERCENTAGE ERROR THRESHOLD ...
          E = (1.0/N)*(SUM(C(I)))                      WHERE I=1,N
               WHERE C(I) = 0         IF ABS((OC(I)-PC(I))/OC(I)).LE.E2
               WHERE C(I) = 1         IF ABS((OC(I)-PC(I))/OC(I)).GT.E2
     5.   SYMMETRIC HIGH-LOW LOSS FUNCTION ...
          E = (1.0/N)*(SUM(C(I)))                      WHERE I=1,N
               WHERE C(I) = 0         IF PC(I) AND OC(I).LE.NORM, E3
                                      IF PC(I) AND OC(I).GT.NORM
               WHERE C(I) = 1         IF OTHERWISE.
     6.   ASYMMETRIC HIGH-LOW LOSS FUNCTION ...
          E = (1.0/N)*(SUM(C(I)))                      WHERE I=1,N
               WHERE C(I) = L1        IF PC(I).GT.NORM, E4
                                      AND OC(I).LT.NORM
               WHERE C(I) = L2        IF PC(I).LT.NORM
                                      AND OC(I).GT.NORM
     7.   USER SUPPLIES LOSS MATRIX ...
          E = (1.0/N)*(SUM(C(I)))                      WHERE I=1,N
               WHERE C(I) = L(J,K)
               WHERE PC(I) = J AND OC(I) = K
     8.   MAXIMUM CONCENTRATION LOCATION ...
          E = (1.0/N)*(SUM(D(I)))                      WHERE I=1,N
                                                       WHERE D(I)=DISTANCE
```

Figure 6. Accuracy score equations.

The program is sufficiently simple that it lends itself to interactive operation, permitting the user to exercise several options, as follows:

- Modify constants within six of the eight tests. (The other two tests involve no such constants.)
- Perform the test for the combined data from all monitoring stations or from each of the stations, or from any selected station.
- Analyze any file in the correct format and order.
- Choose tutorial or brief-prompting interactive modes. (Figure 6 is an example of the summary printed in the tutorial mode.)
- Select any or all the statistical tests.
- Display the accuracy score on a rudimentary map. (For illustrative purposes and subsequent application with the RAPS data base, a St. Louis map was selected.)

Program Logic

The logic of the program structure is illustrated in the flow chart of Figure 7. The FORTRAN V code is given in Section 1 of Appendix C. To simplify relating the flow chart to the source listing, we give the line of the code next to the logic blocks in Figure 4.

Summarizing the logic of Figure 7, we note that the first operator entry is the number of data points and monitoring stations. Knowledge of the number of data points enables us to eliminate computer-specific end-of-file checks or the need for inserting special identifiers indicating the end of the data. The number of stations is obviously well known. The number of data points will be known after prescreening the data. (The SPSS data-display software will list the size of the data base.) The next interaction requires the user to select whether the tests will be conducted on the entire data base, or for all or any one of the monitoring stations. If tests are to be done for each station, the user has the option of displaying the results on a geographical map, assuming a Tektronix 4014 terminal is available.

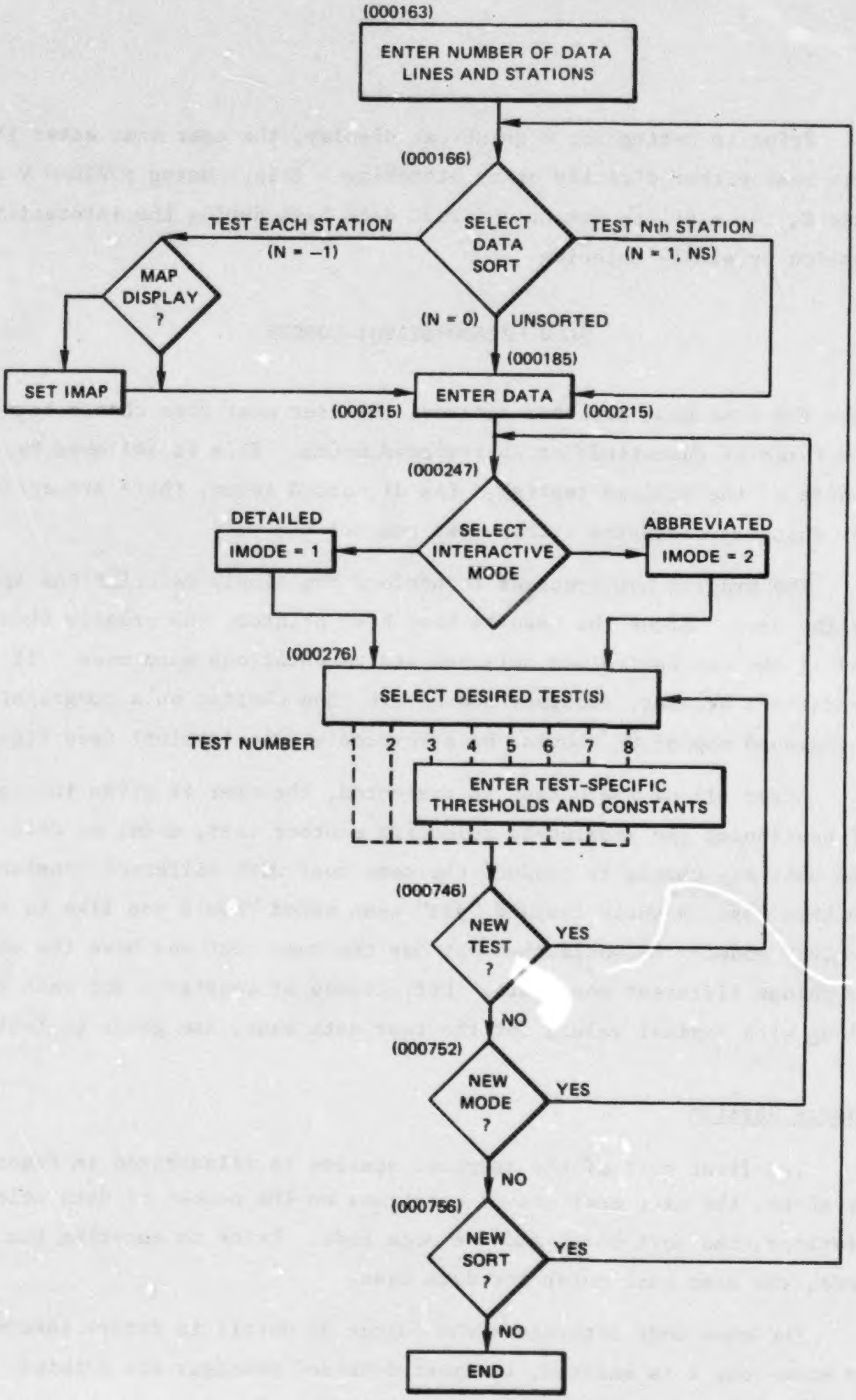

(000163)
ENTER NUMBER OF DATA LINES AND STATIONS
(000166)
SELECT DATA SORT
TEST EACH STATION
(N = −1)
TEST Nth STATION
(N = 1, NS)
MAP DISPLAY ?
SET IMAP
(N = 0)
UNSORTED
(000185)
ENTER DATA
(000215)
(000215)
(000247)
DETAILED
IMODE = 1
SELECT INTERACTIVE MODE
ABBREVIATED
IMODE = 2
(000276)
SELECT DESIRED TEST(S)
TEST NUMBER
1
2
3
4
5
6
7
8
ENTER TEST-SPECIFIC THRESHOLDS AND CONSTANTS
(000746)
NEW TEST ?
YES
NO
(000752)
NEW MODE ?
YES
NO
(000756)
NEW SORT ?
YES
NO
END

Figure 7. Flow diagram for the accuracy-score program.

Prior to opting for a graphical display, the user must enter the data base either directly or by attaching a file. Using FORTRAN V and Exec 8, the user can attach the test data base during the interactive session by simply entering

```
@ADD M21ADO*STAT01.COQS2N   .
```

Once the data base has been entered, the user must then choose between the tutorial (detailed) or abbreviated modes. This is followed by a choice of the desired test(s). (As discussed later, there are options for changing constants within most tests.)

The program now proceeds to perform the simple calculations specified by the user. After the results have been printed, the program checks to see if the map option was selected and computations were made. If these conditions are met, accuracy scores are then plotted on a geographical background map of St. Louis, on a Textronix 4014 terminal (see Figure 8).

After all of the output is presented, the user is given the option of continuing the session by selecting another test, mode, or data sort. The user may choose to conduct the same test with different constants. In this case he would respond "yes" when asked "Would you like to try another mode?" He would then opt for the same test and have the chance to choose different constants. Definitions of constants for each test, along with typical values for the test data base, are given in Table 1.

Sample Session

The first part of the terminal session is illustrated in Figure 9. As shown, the user must answer questions on the number of data points and stations, the sort code, and the mode code. Prior to entering the mode code, the user must enter the data base.

The mode code determines the degree of detail in future interaction. If mode-code 1 is entered, the most detailed messages are printed. The next message, after mode selection, will be a listing of the tests and

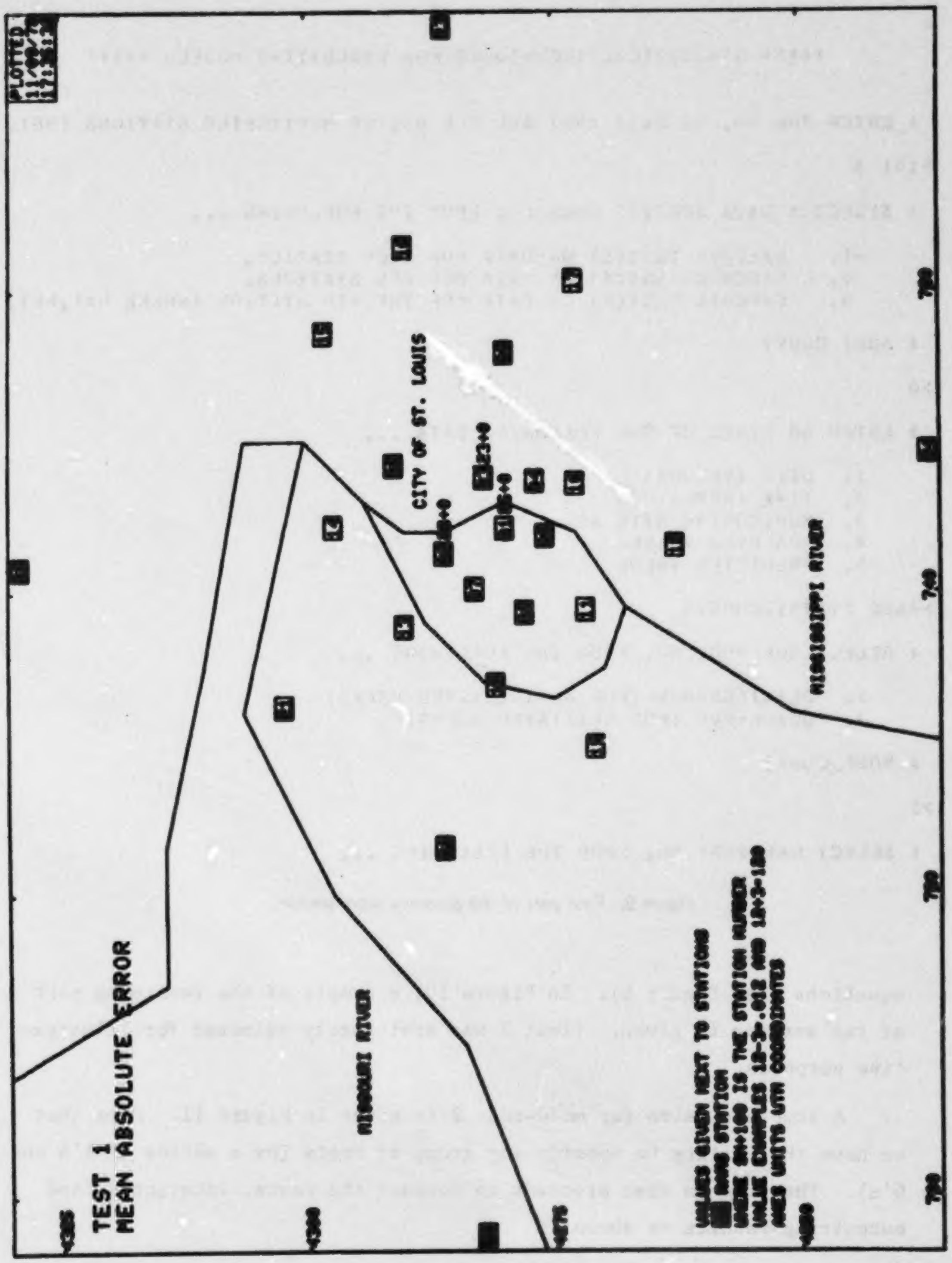
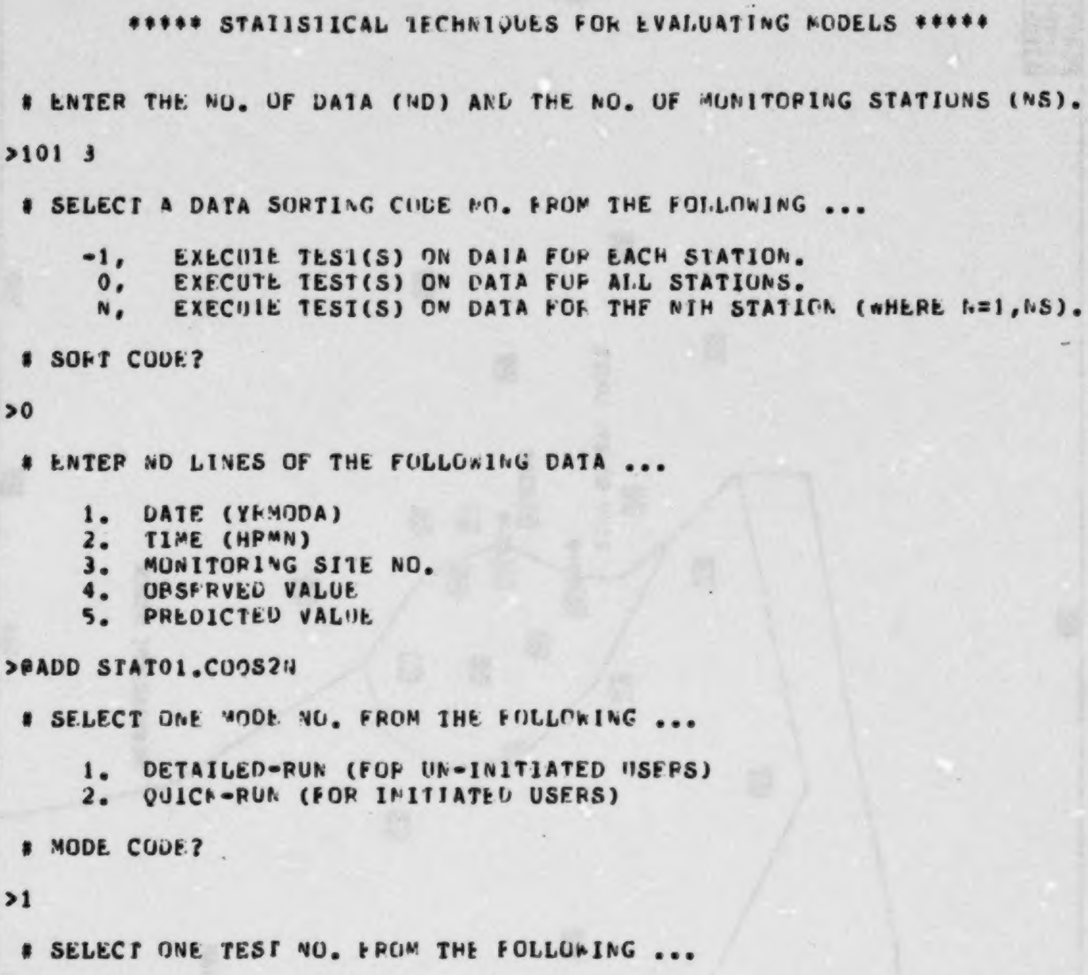

TEST 1
MEAN ABSOLUTE ERROR
CITY OF ST. LOUIS
MISSOURI RIVER
MISSISSIPPI RIVER
VALUES GIVEN NEXT TO STATIONS
RAMS STATION
PLOTTED

Figure 8. Results of Test 1 displayed on a map of St. Louis.



```
***** STATISTICAL TECHNIQUES FOR EVALUATING MODELS *****

 # ENTER THE NO. OF DATA (ND) AND THE NO. OF MONITORING STATIONS (NS).

>101 3

 # SELECT A DATA SORTING CODE NO. FROM THE FOLLOWING ...

     -1,    EXECUTE TEST(S) ON DATA FOR EACH STATION.
      0,    EXECUTE TEST(S) ON DATA FOR ALL STATIONS.
      N,    EXECUTE TEST(S) ON DATA FOR THE NTH STATION (WHERE N=1,NS).

 # SORT CODE?

>0

 # ENTER ND LINES OF THE FOLLOWING DATA ...

     1.   DATE (YRMODA)
     2.   TIME (HRMN)
     3.   MONITORING SITE NO.
     4.   OBSERVED VALUE
     5.   PREDICTED VALUE

>@ADD STAT01.CO0S2H

 # SELECT ONE MODE NO. FROM THE FOLLOWING ...

     1.   DETAILED-RUN (FOR UN-INITIATED USERS)
     2.   QUICK-RUN (FOR INITIATED USERS)

 # MODE CODE?

>1

 # SELECT ONE TEST NO. FROM THE FOLLOWING ...
```

Figure 9. First part of the accuracy score session.

equations (See Figure 6). In Figure 10, a sample of the remaining part of the session is given. (Test 7 was arbitrarily selected for illustrative purposes.

A sample session for mode-code 2 is given in Figure 11. Note that we have the ability to specify any group of tests (by a series of 1's and 0's). The program then proceeds to conduct the tests, interacting and outputting results as shown.

TABLE 1. DEFINITION OF USER-ENTERED VALUES FOR THE ACCURACY SCORE TEST

Test	Constant	Description	Test Value*
3	E1	Error threshold	50
4	E2	Percentage error threshold	0.2
5	E3 (NORM)	Concentration cutoff	400
6	E4 (NORM)	Concentration cutoff	400
	L1	Overprediction loss	1
	L2	Underprediction loss	2
7	--	Loss matrix	STAT01.ERRLOS†
	CMAX	Anticipated maximum concentration	1000
8	--	Distance matrix	STAT01.DISTAN†

*Suggested test values for the test data base, STAT01.COQS2N.

†Test elements in file STAT01.

```
          NOTE ... TEST8 IS VALID ONLY FOR SORT CODE -1.

 # TEST CODE?

>1

   SITE ALL, TEST1: MEAN ABSOLUTE ERROR IS    .458+02

           95% CONFIDENCE INTERVAL:   .384+02 TO    .532+02

 # WOULD YOU LIKE TO TRY ANOTHER TEST ... TYPE YES OR NO.

>NO

 WOULD YOU LIKE TO TRY ANOTHER MODE ... TYPE YES OR NO.

>YES

 # MODE CODE?
```

Figure 10. Accuracy score session (MODE CODE = 1).
See Figure 9 and then Figure 6 for the first part of the session.

```
 #MODE CODE?

>2

 #SELECT TEST NOS. WITH A SERIES OF 1 (YES) AND 0(NO).

     1.  MEAN ABSOLUTE ERROR.
     2.  MEAN SQUARE ERROR.
     3.  ABSOLUTE ERROR THRESHOLD.
     4.  PERCENTAGE ERROR THRESHOLD.
     5.  SYMMETRIC HIGH-LOW LOSS FUNCTION.
     6.  ASYMMETRIC HIGH-LOW LOSS FUNCTION.
     7.  USER SUPPLIED LOSS MATRIX.
     8.  MAXIMUM CONCENTRATION LOCATION.

>1 0 1 0 1 0 1 0

   SITE  01, TEST1:  MEAN ABSOLUTE ERROR IS     .858+02

           85% CONFIDENCE INTERVAL:    .633+02 TO       .109+03

   SITE  02, TEST1:  MEAN ABSOLUTE ERROR IS     .280+02

           95% CONFIDENCE INTERVAL:    .215+02 TO       .345+02

   SITE  03, TEST1:  MEAN ABSOLUTE ERROR IS     .232+02

           95% CONFIDENCE INTERVAL:    .191+02 TO       .272+02

 #TEST3, ENTER E1:

>25

   SITE  01, TEST3:  ABSOLUTE ERROR THRESHOLD IS   .765+00

           95% CONFIDENCE INTERVAL:    .695+00 TO       .835+00

   SITE  02, TEST3:  ABSOLUTE ERROR THRESHOLD IS   .364+00

           95% CONFIDENCE INTERVAL:    .230+00 TO       .497+00

   SITE  03, TEST3:  ABSOLUTE ERROR THRESHOLD IS   .294+00

           95% CONFIDENCE INTERVAL:    .204+00 TO       .385+00

 #TEST5, ENTER E3:
```

Figure 11. Accuracy score session (MODE CODE = 2).

Program Structure

The program (software) structure is given in Figure 12. (The Exec 8 control statements that "map" or link the subroutines are contained in file STAT01.MAPIT. A listing is given in Section 2 of Appendix C.) As shown in Figure 12, the main program, STAT01.ACSCOR, calls for statistical subroutines, STAT01.TESTS and STAT01.CONLIM, and, as a user option, plotting-routine SRI*SRI.RAMMAP. This plotting routine calls two FORTRAN subroutines, SRI*SRI.BOX and SRI*SRI.HEADER, a special RAPS utility, RAPS*UTILITY.COMPOZ, and the standard Tektronix software, GRAPH*TEKTRONIX.

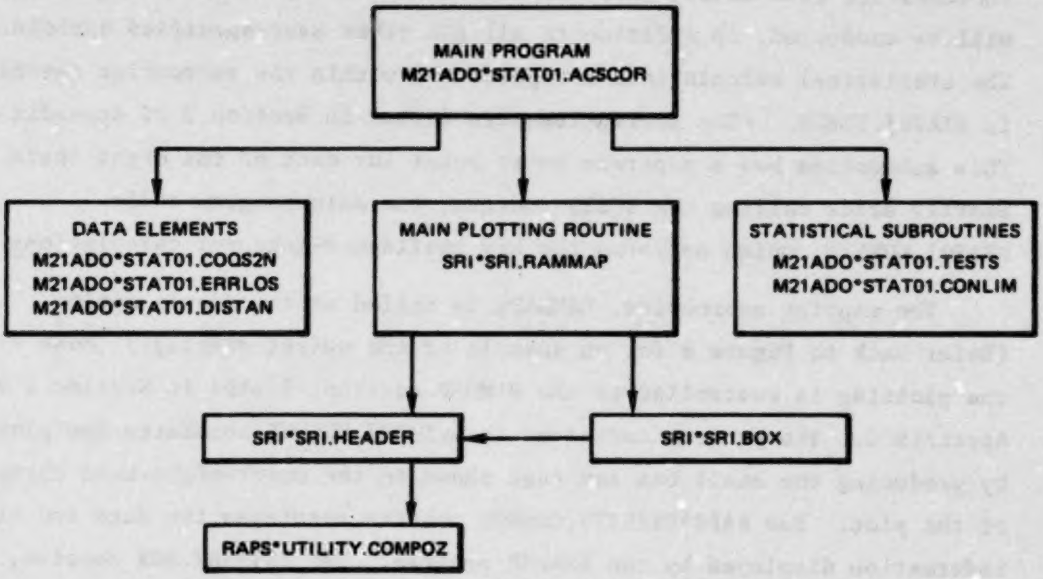

MAIN PROGRAM
M21ADO*STAT01.ACSCOR
DATA ELEMENTS
M21ADO*STAT01.COQS2N
M21ADO*STAT01.ERRLOS
M21ADO*STAT01.DISTAN
MAIN PLOTTING ROUTINE
SRI*SRI.RAMMAP
STATISTICAL SUBROUTINES
M21ADO*STAT01.TESTS
M21ADO*STAT01.CONLIM
SRI*SRI.HEADER
SRI*SRI.BOX
RAPS*UTILITY.COMPOZ

Figure 12. Software structure for the accuracy score program.

The data elements in STAT01 are available as a user option for testing the software. The first, COQS2N, is the test data base previously described. The second element, ERRLOS, can be used in Test 7 as the user-supplied loss matrix, while the third, DISTAN, can be used in Test 8 as the distance-loss matrix. These data elements are further discussed in the next subsection, Data Entry.

The programs are written in the FORTRAN V language. The main program and statistical subroutines are written for general application on any computer. The plotting programs, however, do use some specialized software (e.g., RAPS*UTILITY.COMPOZ). Also, the map is specific to the St. Louis area and the RAPS monitoring locations. Therefore, when applying the accuracy score programs to other geographical areas, one must, of necessity, write new plotting software for the specific area in question.

The primary function of the main program, STAT01.ACSCOR, is to control the interaction between the user and the computer. This interaction includes the data entry, which in turn controls which of the eight tests will be conducted, in addition to all the other user-specified options. The statistical calculations are performed within the subroutine contained in STAT01.TESTS. (The subroutines are listed in Section 3 of Appendix C.) This subroutine has a separate entry point for each of the eight tests. Shortly after calling the TESTS routine, the main program calls STAT01.CONLIM, which performs the key confidence-interval calculations.

The mapping subroutine, RAMMAP, is called at the user's option. (Refer back to Figure 8 for an example of the output display.) Most of the plotting is controlled in the RAMMAP routine, listed in Section 3 of Appendix C. The program contained in SRI*SRI.HEADER annotates the plot by producing the small box and test shown in the upper-right-hand corner of the plot. The RAPS*UTILITY.COMPOZ routine retrieves the date and time information displayed by the HEADER program. The SRI*SRI.BOX routine, called by both RAMMAP and HEADER, is used to draw frames around the map and header. All the above routines are listed in Section 3 of Appendix C.

Data Entry

Some of the information on data entry has just been described. As stated, the user, upon interrogation, must enter the number of data points and monitoring stations. The other interrogations also involve entry of one or two variables, as described in Table 1.

The data base to be tested is requested during the session by the prompt:

```
# ENTER ND LINES OF THE FOLLOWING DATA...
          1.  DATE (YRMODA)
          2.  TIME (HRMN)
          3.  MONITORING SITE NO.
          4.  OBSERVED VALUE
          5.  PREDICTED VALUE
```

The contents and format of the test data base is given in Table 2. (See Appendix A for the listing of the test data base.) While the ISTC, IWSC, and IWDC parameters are not used within the program, we must allow for them when entering our data. Using the first line of data in Appendix A as an example, we could format the first line of our data entry at the terminal as follows (where the three consecutive 2's are arbitrary but necessary for formatting purposes):

```
741012,0900,1,2,2,2,309.7,322.8   .
```

TABLE 2. DESCRIPTION AND FORMAT OF TEST DATA BASE

Item	Name	Format	Description
1	IDATE	I6	Calendar date--YRMODA
2	ITIME	I4	Time, 0000-2300
3	ISITE	I1	Monitoring station number, 1-3
4	ISTC	I1	Stability class, 1-6
5	IWSC	I1	Wind speed class, 1-6
6	IWDC	I2	Wind direction class, 1-16
7	OCS	F5.1	Observed concentration
8	PCS	F5.1	Predicted concentration

The above format would be repeated for each set of comparisons ("ND" lines total). For data entry, the user has the option of attaching a data base in an existing file or entering the data base during the session. In the former case, as previously stated, a data base can be attached by entering a statement such as:

```
@ADD M21ADO*STAT01.COQS2N    ,
```

which will enter our test data base. Alternatively, we can enter data as specified by the "READ" statement (see lines 000185, 000215, and so forth, of the ACSCOR code):

```
READ(INP,3)IDATE(I),ITIME(I),ISITE(I),ISTC,IWSC,IWD,OCS,PCS
```

where the first three parameters (IDATE,ITIME,ISITE) and last two parameters (OCS,PCS) are specified, and dummy variables are entered for ISTC, IWSC, and IWDC.

After data are entered, each record is examined to ensure that the observed and predicted concentrations are valid (or missing). Invalid data are designated as such by negative numbers. The ACSCOR program checks to see if either of the concentrations is negative (e.g., see line 000190). It negative, the data are not used in the calculation.

The remaining input data are entered as shown in the sample interactive sessions (see Figure 9). Usually, each query involves the input of one or two numbers, but there are two exceptions, Test 7 and Test 8.

When Test 7 is specified, the program will request a user-supplied loss matrix. As the program now stands, this is a 5 by 5 loss matrix that can be entered directly during the session or by attaching a file. For instance, our test file (see the listing in Section 4 of Appendix C) can be entered by:

```
@ADD M21ADO*STAT01.ERRLOS    .
```

Alternatively, we could directly enter five data lines, or rows (J = 1 to 5), each with five loss indexes (K = 1 to 5). Each index is a loss associated with predicting concentrations in class J while observing them in class K. (The loss is usually set to zero when J = K).

As shown in line 000627 of the ACSCOR program, the loss matrix is read into the AS array. This corresponds to five evenly divided categories of predicted and observed concentrations. The categories J and K are calculated in lines 000059 and 000061 of the TESTS program. Note that 5 and 0.2 (1/5) determine how many categories are entered. CMAX is the user-entered expected maximum concentration (see Table 1). The number of categories can be respecified by changing lines 000059 through 000062 and the size of the AS array in the ACSCOR and TESTS programs.

For Test 8, the distance-loss matrix is currently specified as a 3 by 3 array because there are three stations. When asked for the distance-loss matrix, the user can attach the existing file (see Section 4 of Appendix C) by entering:

```
@ADD M21ADO*STAT01.DISTAN   .
```

Alternatively, the user can enter three rows (J = 1 to 3), each containing three loss indexes (K = 1 to 3). The resulting matrix should be symmetric. Each index in the matrix is a loss associated with predicting the maximum concentration at one location, J, while observing it at another, K. (The loss is usually set to zero when J = K.) This information is read into the DD array (see line 000703 of the ACSCOR program). The number of monitoring stations previously entered determines the required size of the entered matrix. The DD array needs to be dimensioned so that the number of rows and columns equals the maximum number of monitoring stations. (Even though a 3 by 3 matrix is sufficient, the DD array is currently dimensioned by a 5 by 5 matrix in the COMMON statements labeled DAT.) The above changes are all that are required to adapt the program for inclusion of additional stations.

RESIDUAL TIME SERIES (STATO2.TIMSER)

The residual time series program is an "intermediate evaluation statistic" used to detect cyclical components in the data. An unbroken sequence of observed-predicted concentration pairs is required. Missing concentration values are accepted by the program if flagged in sequence with a negative number. The output is entirely graphic, so it is assumed that a Tektronix 4014 terminal will be used. There are two graphs displayed for each station. The first, shown in Figure 13, is a plot of the autocorrelation function against time lag (in user units which equal the concentration averaging time). The second, shown in Figure 14, is a plot of the cumulative periodogram. These plots are generated for each monitoring station of interest.

From a programming standpoint, the logic is quite simple because it contains few options. The program structure for the Residual Time Series Program is shown in Figure 15. Execution of the program and data entry are controlled by the Control Element, M21ADO*STATO2.CASE1. (Parameters within the Control Element may be varied to specify different options or data. Hence, we refer to CASEn in Figure 7 because there are many variations. However, the structure remains the same.) As shown in the listing for CASE1, Section 1 of Appendix D, the first line specifies the execution of the program contained in M21ADO*STATO2.TIMSER. Execution for CASE1 is begun by entering on the Tektronix terminal:

```
@ADD,L M21ADO*STATO2.CASE1   .
```

Execution of the TIMSER program will then begin, with the data beginning in line 000002 being read in under program control.

The mapping of the subroutines required by the TIMSER program is shown in Section 2 of Appendix D. The source listing for the main program, TIMSER, is given in Section 3 of Appendix D. The main program performs all the calculations, most of the plotting, and the data entry.

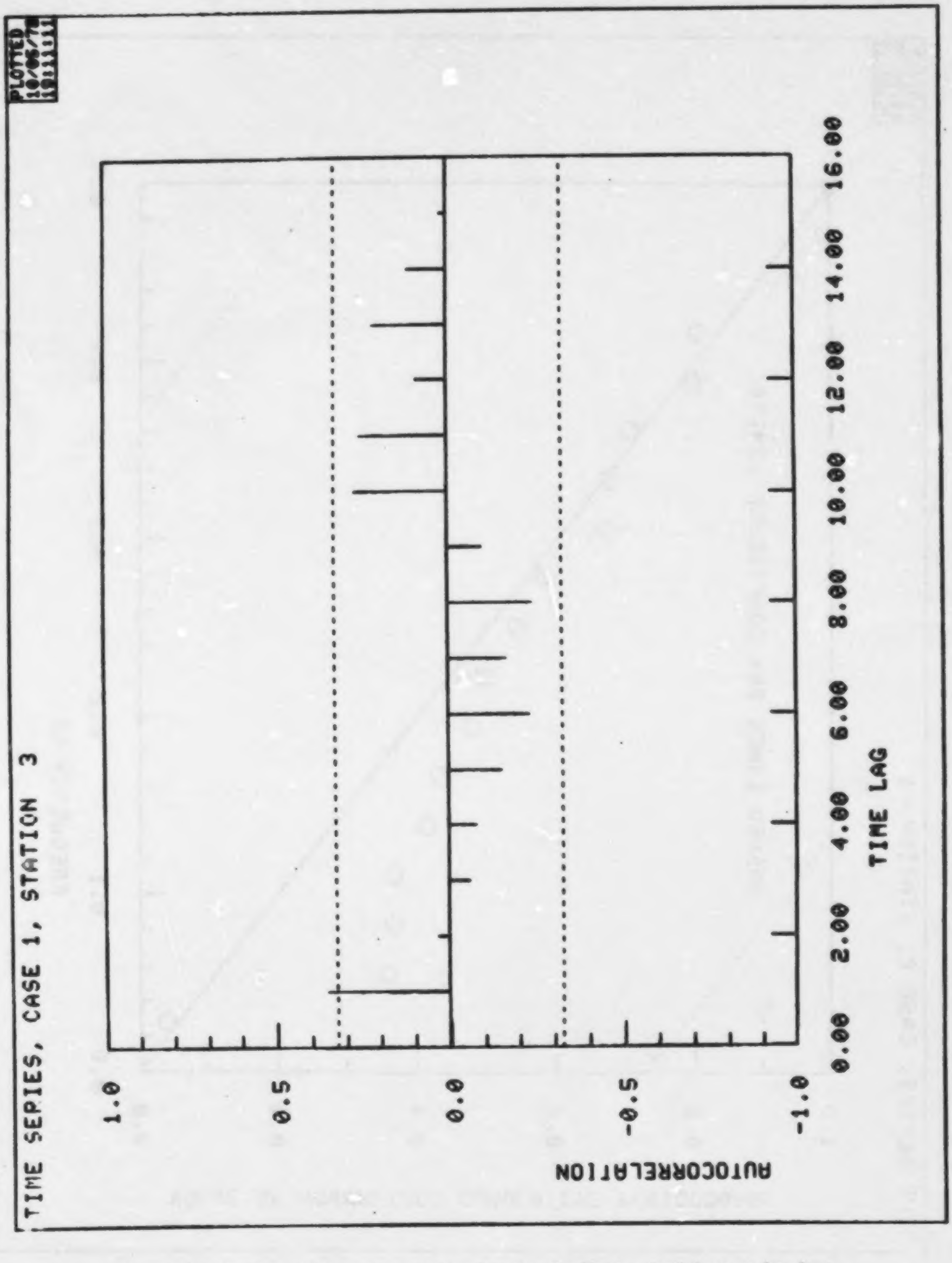


Figure 13. Display of the autocorrelation function for Station 1 of the test data base.

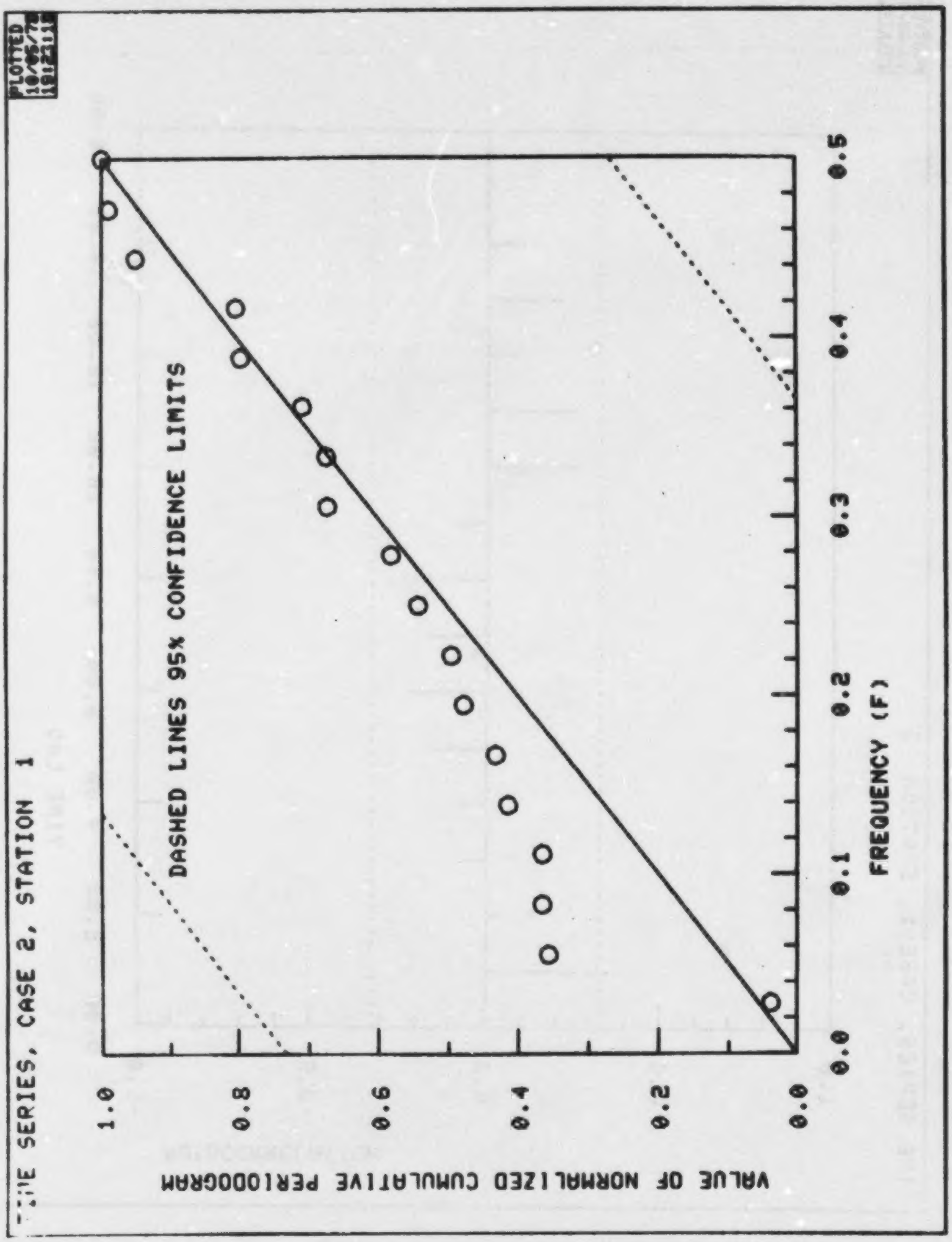


Figure 14. Display of the normalized cumulation periodogram for Station 1 of the test data base.

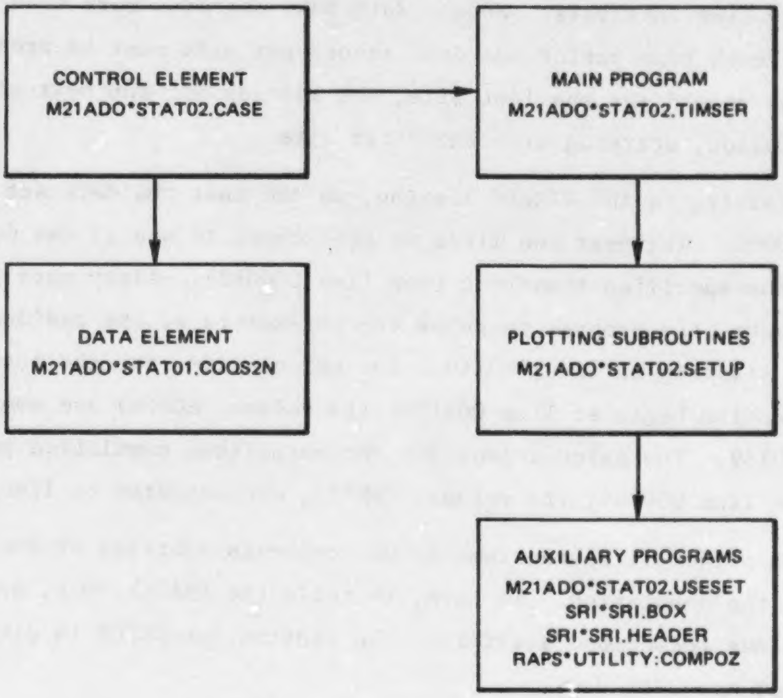

CONTROL ELEMENT
M21ADO*STAT02.CASE
MAIN PROGRAM
M21ADO*STAT02.TIMSER
DATA ELEMENT
M21ADO*STAT01.COQS2N
PLOTTING SUBROUTINES
M21ADO*STAT02.SETUP
AUXILIARY PROGRAMS
M21ADO*STAT02.USESET
SRI*SRI.BOX
SRI*SRI.HEADER
RAPS*UTILITY:COMPOZ

Figure 15. Program structure for the residual time series.

Referring to the TIMSER listing, we see that comment statements (lines 000041 through 000081) are used to define the input data requirements and program options. The corresponding READ statements are immediately before and after the comment statements. The input data will be read line by line, starting with line 000002 of the Control Element, Section 1 of Appendix D. The data base for observed and predicted concentration is specified in line 000011, our COQS2N test data base. Alternatively, other elements could be specified or, as discussed earlier on page 27, the data could be specified directly, starting in line 000011 and continuing for ND lines.

The format requirements of the TIMSER program for data input are identical to those of the other programs. However, the time sequence of the data is important. In general, the data should be read in equal, ascending time intervals. When a data base contains more than one site, then for each time period one data record per site must be presented. After the record for the last site, the records for the next time period should follow, starting with the first site.

Referring to the TIMSER listing, we see that the data are read on line 000092. The next two lines of code check to see if the data fall within the specified timeframe (see line 000082). After more time and data checks, the program computes the parameters of the residual time series, starting in line 000119. The calculations for the autocorrelation function begin at line 000121; its values, ACF(K) are computed on line 000159. The calculations for the normalized cumulative periodogram begin on line 000209; its values, CNP(J), are computed on line 000231.

The principal subroutine, SETUP, controls plotting of the axis and some of the annotation. In turn, it calls the HEADER, BOX, and COMPOZ subroutines (described earlier). The listing for SETUP is given in Section 4 of Appendix D.

CHI-SQUARE GOODNESS-OF-FIT (STAT03.CHIFIT)

The chi-square goodness-of-fit program can be useful as either an intermediate or a final evaluation statistic. As was the case for the residual time series program, the output is completely graphic. So a Tektronix 4014 terminal must be used. The output consists of two displays, each containing two plots. The first output display, Figure 16, consists of histograms for both the observed and predicted concentration data sets; the second output display, Figure 17, consists of histograms of the difference between observed and predicted concentrations, absolute and percentage.

The chi-square statistic, X^2, and the number of samples, N, are printed in the lower-right-hand corner of the second plot. (The statistic, X^2, is simply the summation of the square of the observed and predicted concentration differences divided by the observed concentration.)

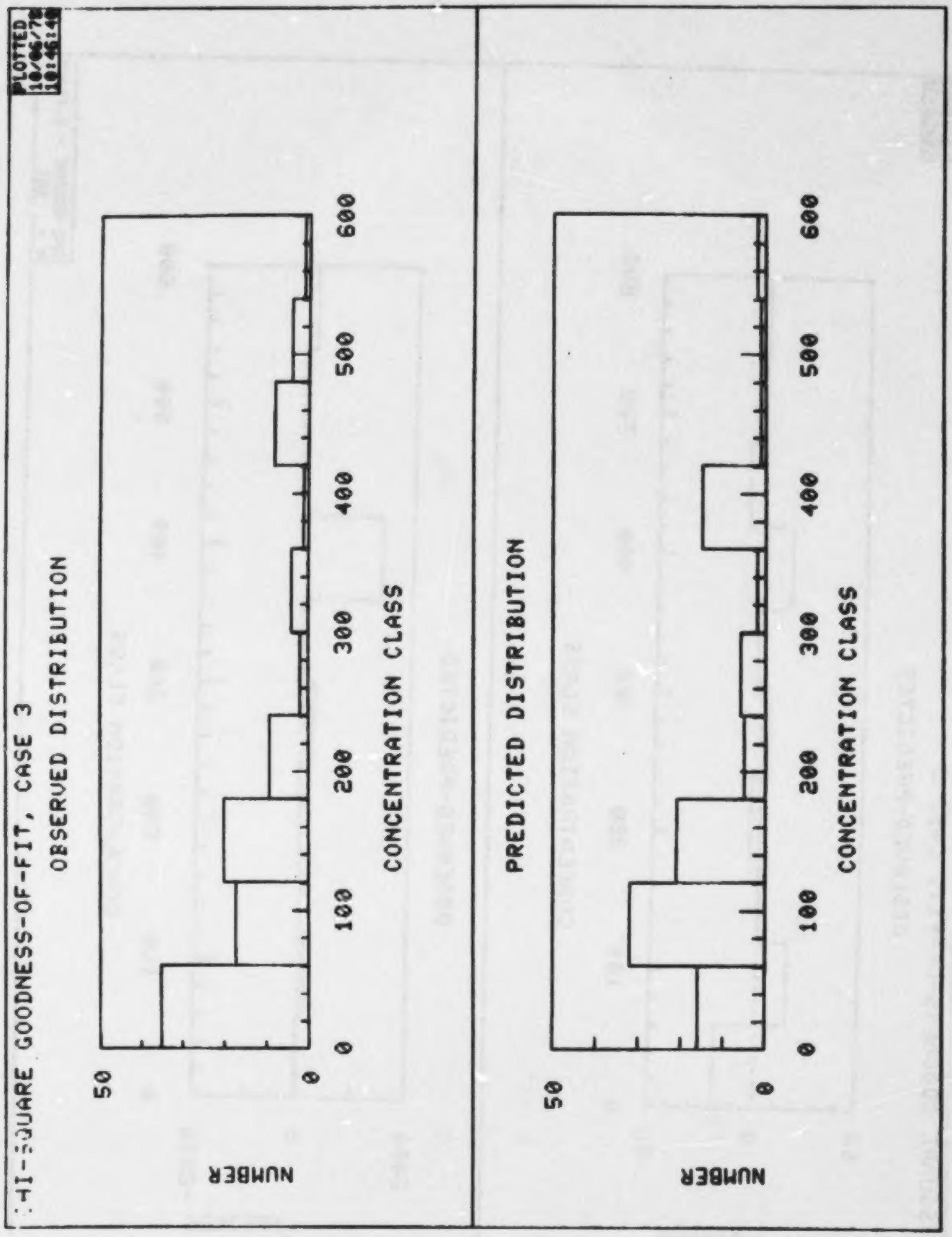


Figure 16. Histogram of concentrations for the test data base.

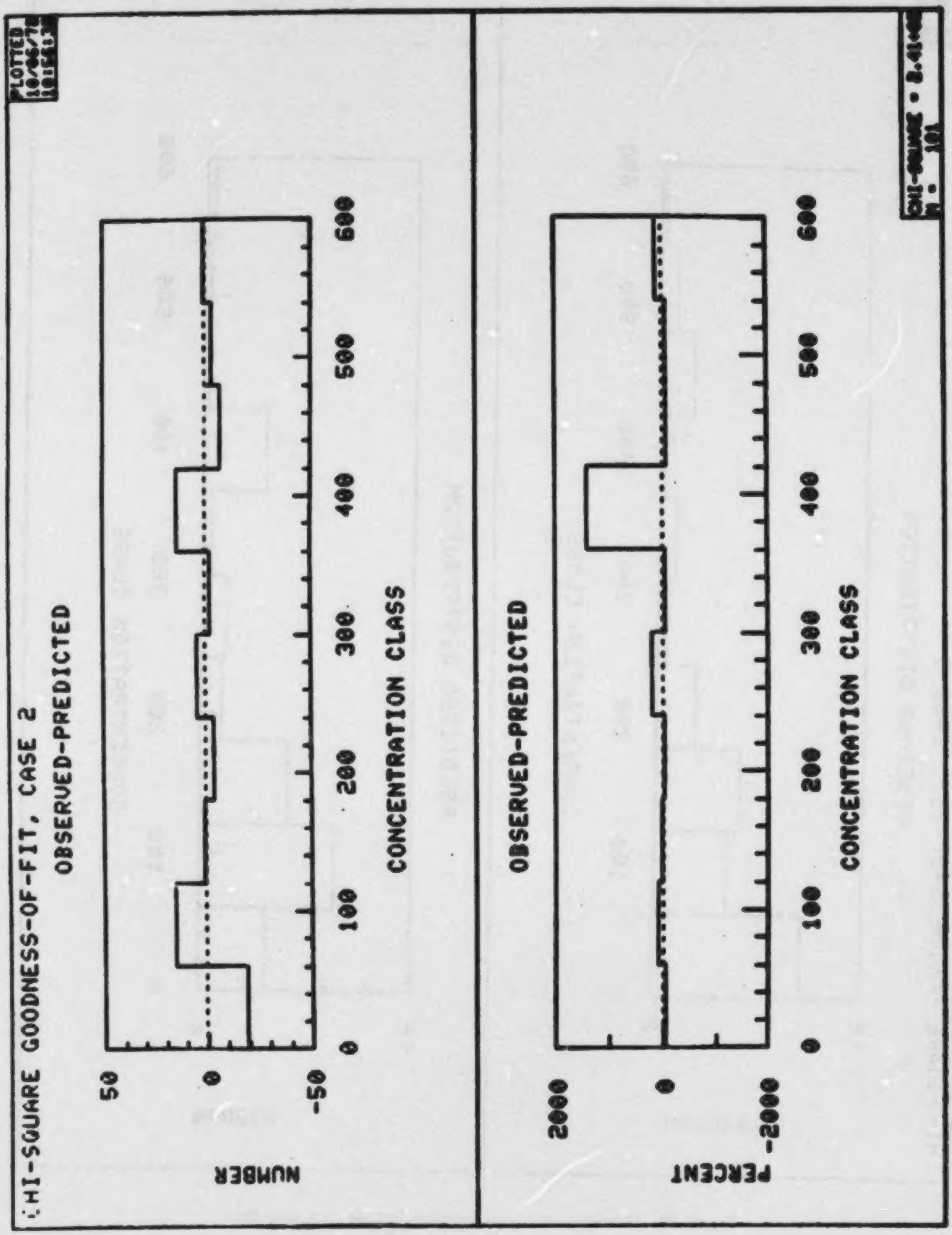


Figure 17. Histogram of concentration residuals for the test data base.

The program structure for the chi-square program is shown in Figure 18. The Control Element is used to execute the program and supply the requisite input data. To initiate the program from the Tektronix terminal, the user gives the Exec 8 command (see listings for the Control Element in Section 1 of Appendix E):

```
@ADD,L M21ADO*STAT03.CASE1   .
```

Note that the first line specifies execution of the program M21ADO*STAT03.CHIFIT. Its elements are shown in the mapping routine, listed in Section 2 of Appendix E.

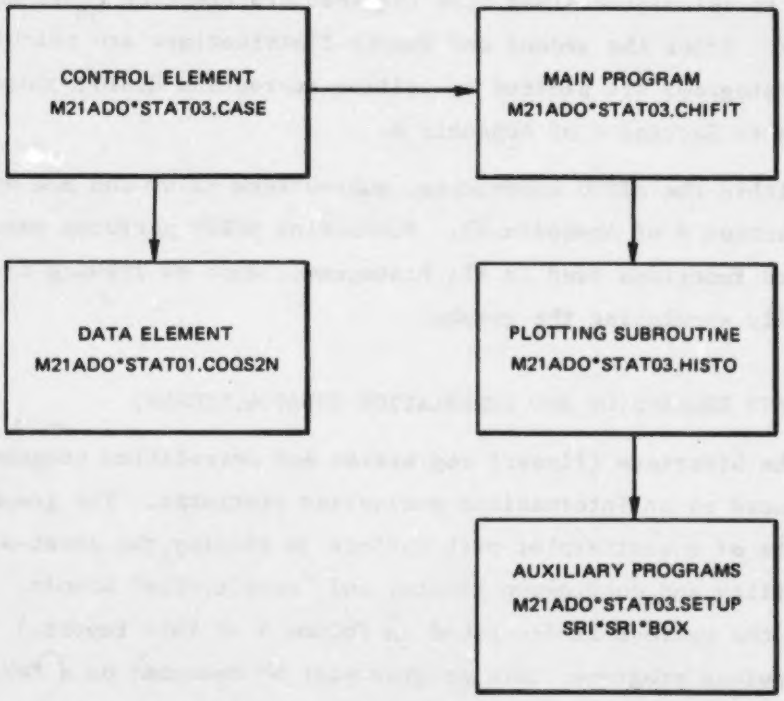

CONTROL ELEMENT
M21ADO*STAT03.CASE
MAIN PROGRAM
M21ADO*STAT03.CHIFIT
DATA ELEMENT
M21ADO*STAT01.COQS2N
PLOTTING SUBROUTINE
M21ADO*STAT03.HISTO
AUXILIARY PROGRAMS
M21ADO*STAT03.SETUP
SRI*SRI*BOX

Figure 18. Structure of the chi-square goodness-of-fit program.

The main program, CHIFIT, controls the data flow, computes χ^2, divides the data into categories, and computes the frequency of occurrence in each category. The listing for CHIFIT is given in Section 3 of Appendix E. The input data for CASE1 starts in line 2 (Section 1, Appendix E). Each line corresponds to a READ statement, the first of which appears as line 000043 of CHIFIT. The last READ statement (line 000073) specifies the entry of the concentration data base. (For CASE1, our test data base STAT01.COQS2N is specified.) Each read-in parameter is defined in the listing for CHIFIT (lines 000023 through 000042 and lines 000055 through 000066).

The four frequency distributions are calculated within CHIFIT, beginning in lines 000087, 000113, 000132, and 000142. The χ^2 statistic is calculated along with the last distribution (CHI2 in line 000148). After the second and fourth distributions are calculated, the dual histograms are plotted by calling subroutine HISTO, whose listing appears in Section 4 of Appendix E.

Within the HISTO subroutine, subroutines SETUP and BOX are called (see Section 4 of Appendix E). Subroutine SETUP performs some of the standard functions used in all histograms, such as framing the plots and partially annotating the graphs.

BIVARIATE REGRESSION AND CORRELATION (STAT04.REGANA)

The bivariate (linear) regression and correlation program is most often used as an intermediate evaluation statistic. The graphic output consists of a scatterplot with options to display the least-squares fit, probability and confidence limits, and "sensitivity" bounds. (The utility of the options is described in Volume 4 of this report.) As for the two previous programs, this program must be executed on a Tektronix 4014 terminal. An example of the program output is given in Figure 19. Note that the correlation coefficient, regression coefficients, and number of samples are displayed in the lower-left-hand corner of the plot.

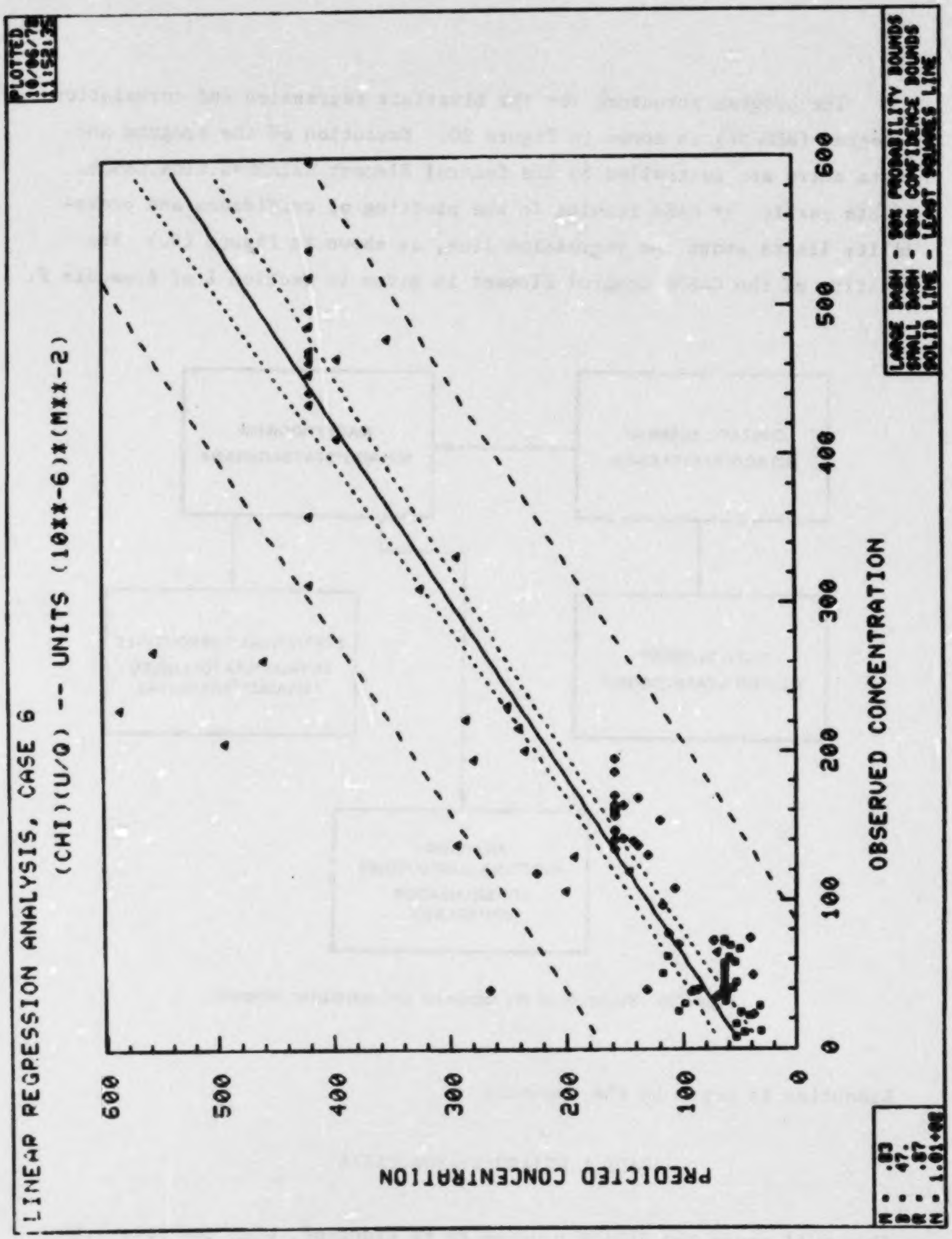


Figure 19. Scattergram with least-squares, confidence, and probability lines.

The program structure for the bivariate regression and correlation program (REGANA) is shown in Figure 20. Execution of the program and data entry are controlled by the Control Element M21ADO*STAT04.CASE6. (This particular CASE results in the plotting of confidence and probability limits about the regression line, as shown in Figure 19.) The listing of the CASE6 Control Element is given in Section 1 of Appendix F.

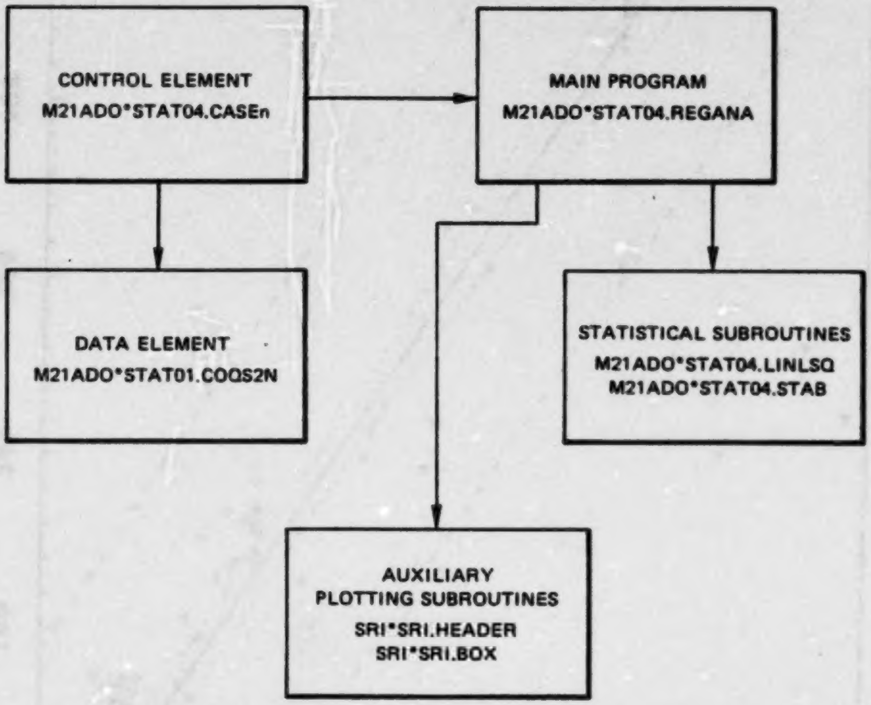

CONTROL ELEMENT
M21ADO*STAT04.CASEn
MAIN PROGRAM
M21ADO*STAT04.REGANA
DATA ELEMENT
M21ADO*STAT01.COQS2N
STATISTICAL SUBROUTINES
M21ADO*STAT04.LINLSQ
M21ADO*STAT04.STAB
AUXILIARY
PLOTTING SUBROUTINES
SRI*SRI.HEADER
SRI*SRI.BOX

Figure 20. Structure of the regression and correlation program.

Execution is begun by the command:

```
@ADD,L M21ADO*STAT04.CASE6   .
```

This will cause the REGANA program to be executed, with the data beginning in line 000002 being read in under program control.

The mapping of the subroutines required by the REGANA is shown in Section 2 of Appendix F. The source listing for the main program is given in Section 3 of Appendix F. The main program controls the entering of data, calling of statistical subroutines, and plotting of scattergrams, regression lines, and confidence limits.

Referring to the REGANA listing, we see that comment statements (lines 000023 through 000045 and lines 000060 and 000071) are used to define the input data requirements and program options. The corresponding READ statements start in line 000046 and culminate in line 000076. (These two lines correspond to the data given in lines 000002 and 000017 in the CASE6 element.) In line 000103, REGANA calls the LINLSQ subroutine, which computes all the statistical parameters of interest. The program then proceeds to plot the scattergram (line 000138) and limit lines, depending on the options (the ICODE parameters).

In plotting the scattergram, the program distinguishes among the site codes (ISITE) for each of the three stations in the test data base. They are stored in three separate arrays--(X1, Y1), (X2, Y2), and (X3, Y3)--and plotted with different symbols. The logic starting in line 000081 and ending in line 000096 determines that the array suffix corresponds to the monitoring station number. The actual plotting of the arrays is programmed in lines 000216, 000224, and 000228.

The parameter ICODE determines which, if any, of the regression or limit lines will be plotted on the scattergram. The READ statement for ICODE is in line 000048. (For our example, the data are contained in line 000003 of the CASE6 element. The first parameter, "6", is ICODE for this case. The second parameter specifies that the X and Y axes will be linear.) The logic starting in line 000232 shows how the ICODE value controls the six different options available to the user.

The ICODE options are defined in the comment statements (lines 000027 through 000033). Specification of ICODEs of 3 through 6 causes statistical subroutines to be called. Subroutine LINLSQ, which is called for all ICODEs, is used somewhat differently when an ICODE of 3 is specified. Note that after sensitivity bounds are calculated (lines 000116 to 000124), the LINLSQ program is called (line 000134) for only those points falling outside the sensitivity bounds. (Note the sensitivity bound check in line 000129.) The significance of this calculation is discussed in Volume 4.

For ICODEs of 4 through 6, the FORTRAN function routine, STAB, is called in line 000256. (See Section 4 of Appendix F for a listing of STAB.) STAB assists in computing the 90 percent confidence intervals using Student's t-distribution read into the TAB array.

The subroutine LINLSQ, called in line 000103, computes most of the statistical parameters of interest. The listing for LINLSQ is given in Section 4 of Appendix F. The label COMMON LSQ contains most of the parameters of interest, as follows:

RECNOP - reciprocal of the number of samples
A - constant in the regression-line equation
B - X coefficient in the regression-line equation
XSD - standard deviation of observed concentrations
YSD - standard deviation of predicted concentrations
CIL - lower 95 percent confidence limit for the correlation coefficient
CIU - upper 95 percent confidence limit for the correlation coefficient
SXMXB - square of the accumulated differences between the actual and mean observed concentrations
RC - estimate of Pearson's Correlation Coefficient
RMSE - root-mean-square error
ESD - estimated standard deviation of predicted concentrations about the regression limit
XBAR - average observed concentration.

Not all the above parameters are used in subsequent calculations by the main program REGANA. However, they are useful statistical parameters that can be displayed as an option.

INTERSTATION ERROR CORRELATION (STATO5.COREL2)

The interstation error correlation (COREL2) test computes the correlation coefficient between the residual concentration (observed minus predicted concentrations) at different monitoring stations. The output consists of two matricies. The sample output for our test data base, COQS2N, is shown in Figure 21. The first matrix is the correlation coefficient between each of the true stations in our data base. The second matrix shows the upper and lower 95 percent confidence limits for the correlation coefficients.

```
CORRELATION MATRIX ...

SITE       1       2       3

  1      1.00     .78     .75

  2       .78    1.00     .74

  3       .75     .74    1.00

95-PERCENT CONFIDENCE LIMITS MATRIX ...

SITE       1       2       3

  1 L    1.00     .60     .54
    U    1.00     .89     .87

  2 L     .60    1.00     .52
    U     .89    1.00     .87

  3 L     .54     .52    1.00
    U     .87     .87    1.00
```

Figure 21. COREL2 output for test data base.

Figure 22 displays the structure of the COREL2 program, whose subroutines are shown mapped in the listing contained in Section 1 of Appendix G. The main COREL2 program (listed in Section 2 of Appendix G) controls data input and output and calls the statistical subroutine XYCORR, which controls the calculation of the correlation coefficient.

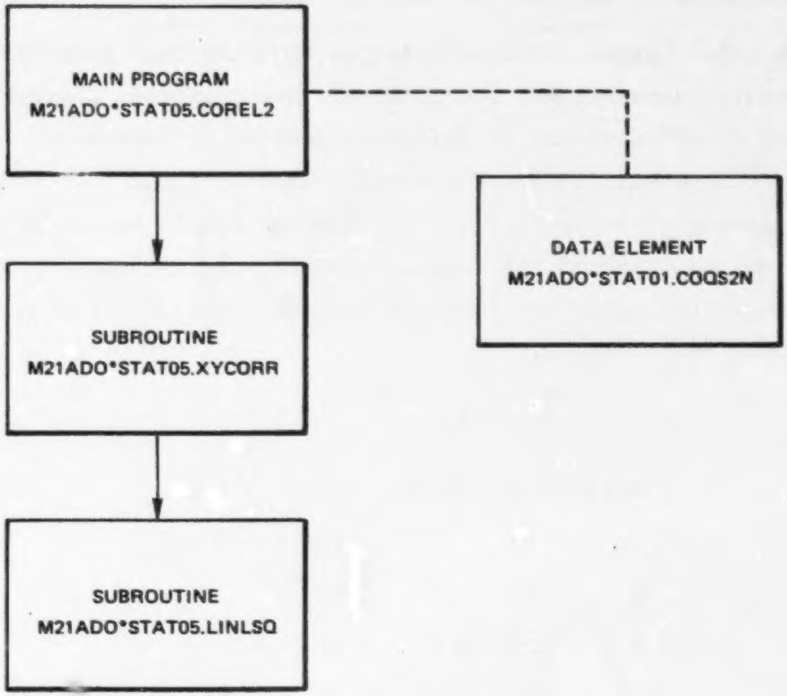

MAIN PROGRAM
M21ADO*STAT05.COREL2
DATA ELEMENT
M21ADO*STAT01.COQS2N
SUBROUTINE
M21ADO*STAT05.XYCORR
SUBROUTINE
M21ADO*STAT05.LINLSQ

Figure 22. Structure of the interstation error correlation program.

The input parameters (parameters are defined in lines 000016 through 000030) are entered as specified by the READ statements in lines 000033 and 000046 of the main program. The first input requires the number of data points, the number of stations, the option code, and the first and second monitoring station numbers. For our test data base, if we want the comparisons for all three stations, the following input would be entered:

```
101,3,2,1,2   .
```

The next read statement requires 101 lines of input data in the prescribed order. Using our test data base, we would enter:

```
@ADD M21ADO*STAT01.COQS2N   .
```

Entering our input data in this order will produce the output shown in Figure 21.

The subroutine stored in M21ADO*STAT05.XYCORR computes the residual concentrations for each monitoring station. As shown in the listing (Section 3 of Appendix G), XYCORR computes the DC1 and DC2 arrays (lines 000013 and 000014) and then calls the LINLSQ subroutine (line 000019). The LINLSQ subroutine, listed in Section 4 of Appendix F, is the same one used in the STAT04.REGANA program.

MULTIPLE REGRESSION OF ERROR RESIDUALS (STAT06.SPSSRUN)

The multiple regression of error residuals, using the meteorological and emission classes as the independent variables, is an intermediate evaluation statistic. Its purpose is to detect the model input parameter that contributes most to the difference between observed and predicted concentrations. These residuals were computed in the STAT06.SPSSDAT program described in Section 2.

The procedure for conducting a regression analysis on the residuals in SPSSFILE is given in Figure 23.

The general format of an SPSS program was described earlier. The program simply finds the correct file, SPSSDATA, which is contained in SPSSFILE. (Using Exec 8, the program automatically looks to Unit 3 for the desired file.) Three specifications are found in the field of the SPSS REGRESSION statement. The first, VARIABLES, identifies the parameters required. If any are missing, a diagnostic will be given. The second and third specifications are REGRESSIONs. Each of the REGRESSION statements identifies the type of regression to be performed. The first, RESID with ASC, WSC, WDC (1), specifies the regression of the concentration residuals on the three meteorological parameters; the second

```
@DELETE PRINT2.
@ASG,PU PRINT2.
@ASG,A SPSSFILE.
@USE 3.,SPSSFILE.
@BRKPT PRINT$/PRINT2
@SPSS
RUN NAME        TEST RUN NO. 1
GET FILE        SPSSDATA
REGRESSION      VARIABLES = TIME,SITE,ASC,WSC,WDC,RESID/
                REGRESSION = RESID WITH ASC,WSC,WDC(1)/
                REGRESSION = RESID WITH TIME,SITE(1)/
OPTIONS         6,11
STATISTICS      ALL
FINISH
@BRKPT PRINT$
@FREE PRINT2.
@SYM,U PRINT2.,,FD04PR
```

Figure 23. Example of a regression procedure.

specifies the regression of residuals on time and site. The program output summaries are given in Figure 24 and 25.

The OPTIONS and STATISTICS statements are used to specify which parts of the computations or output presentations are to be included. The reader is referred to the SPSS manual for a complete summary of the features associated with the REGRESSION procedure. Volume 4 of this final report contains a detailed explanation of the interpretations of the program output.

To complement the REGRESSION output, it is sometimes desirable to plot a scattergram. For instance, the regression analysis of residuals on meteorological parameters reveals that the best linear correlation exists between RESID and ASC. SPSS can be used to plot a scattergram on the line printer with three simple statements. With the same Exec 8 control statements given in Figure 23, the following three SPSS statements follow the @SPSS:

```
GET FILE       SPSSDATA
SCATTERGRAM    RESID (LOWEST, HIGHEST) WITH ASC (3, 6)
FINISH
```

```
TEST RUN NO. 1                                                    06 OCT 78     PAGE    6

FILE    SPSSDATA (CREATION DATE = 04 OCT 78)

* * * * * * * * * * * * * * * * * * * * * *  M U L T I P L E   R E G R E S S I O N  * * * * * * * * * * * *  VARIABLE LIST  1
                                                                                                          REGRESSION LIST  1
DEPENDENT VARIABLE..    RESID     RESIDUAL

                                            SUMMARY TABLE

VARIABLE                                MULTIPLE R   R SQUARE   RSQ CHANGE   SIMPLE R            B              BETA

ASC                                       .62443      .38991     .38991      -.62443      -125.26599        -.63949
WDC                                       .62556      .39133     .00142      -.35899         5.20573         .04157
WSC                                       .62586      .39170     .00037       .32352         7.23051         .02257
(CONSTANT)                                                                                  487.88769
```

Figure 24. Summary output for regression of RESID on ASC, WDC, and WSC.

```
TEST RUN NO. 1                                                    06 OCT 78     PAGE    8

FILE    SPSSDATA (CREATION DATE = 04 OCT 78)

* * * * * * * * * * * * * * * * * * * * * *  M U L T I P L E   R E G R E S S I O N  * * * * * * * * * * * *  VARIABLE LIST  1
                                                                                                          REGRESSION LIST  2
DEPENDENT VARIABLE..    RESID     RESIDUAL

                                            SUMMARY TABLE

VARIABLE                                MULTIPLE R   R SQUARE   RSQ CHANGE   SIMPLE R            B              BETA

TIME                                      .26672      .07114     .07114      -.26672        -.04273         -.26618
SITE                                      .26753      .07157     .00043       .02772        1.90400          .02084
(CONSTANT)                                                                                  20.95030
```

Figure 25. Summary output for regression of RESID on TIME and SITE.

The output for the SCATTERGRAM procedure is given in Figure 26. Note, as specified in the SCATTERGRAM field, that the RESID scale adjusts to the values between the lowest and highest in the file while the ASC scale is from 3 to 6. Again, refer to the SPSS manual for a complete description of the SCATTERGRAM procedure.

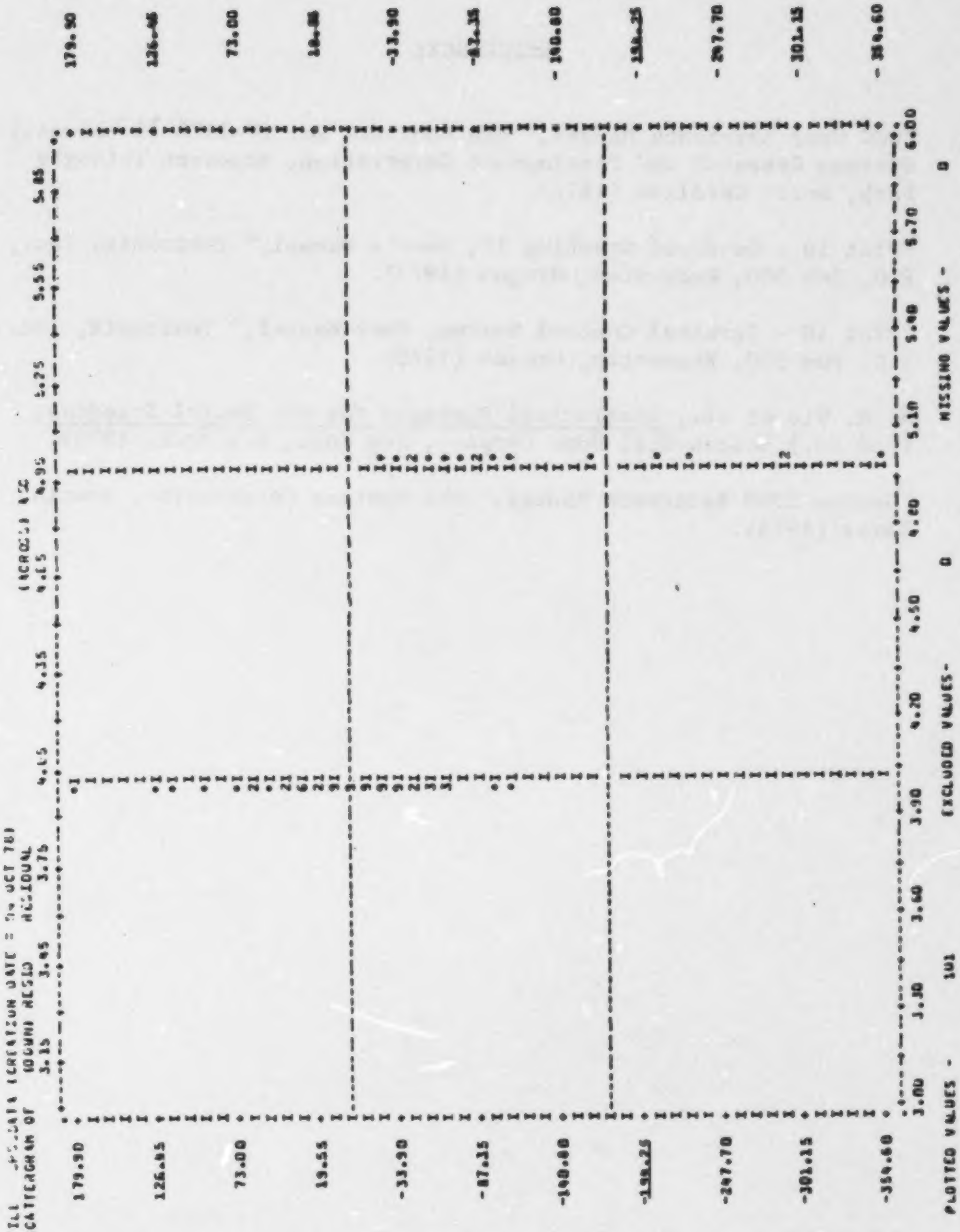


Figure 26. Example of SPSS SCATTERGRAM output.

REFERENCES

1. "NCC User Reference Manual," EPA Contract No. 68-2378 (9 Volumes) Systems Research and Development Corporation, Research Triangle Park, North Carolina (1977).

2. "Plot 10 - Advanced Graphing II, User's Manual," Tektronix, Inc., P.O. Box 500, Beaverton, Oregon (1975).

3. "Plot 10 - Terminal Control System, User Manual," Tektronix, Inc., P.O. Box 500, Beaverton, Oregon (1978).

4. N. H. Nie et al., Statistical Packages for the Social Sciences, (2nd Ed.) McGraw-Hill Book Company, New York, New York, 1975).

5. "System 2000 Reference Manual," MRI Systems Corporation, Austin, Texas (1973).

APPENDIX A

LISTING OF TEST DATA BASE M21ADO*STAT01.COQS2N

```
@ELT,L S1.COQS2N
ELT007 SL73R1 11/01/78 09:38:12 (0,)
C00001        000    741012  0900   1  4  2   2              309.7   322.8
C00002        000    741012  0900   2  4  2   2              130.4   129.1
C00003        000    741012  0900   3  4  2   2               40.6    56.6
C00004        000    741012  1000   1  4  2   2              230.3   248.5
C00005        000    741012  1000   2  4  2   2              107.9   106.2
C00006        000    741012  1000   3  4  2   2               66.8    50.4
C00007        000    741012  1100   1  4  2   2              216.1   239.1
C00008        000    741012  1100   2  4  2   2               70.2   103.2
C00009        000    741012  1100   3  4  2   2               24.9    49.5
C00010        000    741012  1200   1  4  2   3              221.4   283.7
C00011        000    741012  1200   2  4  2   3               97.0   117.1
C00012        000    741012  1200   3  4  2   3               44.4    53.3
C00013        000    741014  1000   1  4  2   1              106.6   198.8
C00014        000    741014  1000   2  4  2   1               39.8    86.6
C00015        000    741014  1000   3  4  2   1               11.1    45.5
C00016        000    741014  1100   1  4  2   1              129.3   192.2
C00017        000    741014  1100   2  4  2   1               38.8    90.4
C00018        000    741014  1100   3  4  2   1                8.4    53.1
C00019        000    741015  0900   1  5  2   3              229.8   584.4
C00020        000    741015  0900   2  5  2   3               40.3   264.7
C00021        000    741015  0900   3  5  2   3               51.6   117.3
C00022        000    741015  1000   1  5  2   3              262.2   457.3
C00023        000    741015  1000   3  5  2   3               30.0    97.6
C00024        000    741015  1100   1  5  2   3              269.0   553.6
C00025        000    741015  1100   3  5  2   3               62.0   112.6
C00026        000    741025  0800   1  4  2   2              477.3   350.8
C00027        000    741025  0800   2  4  2   2              168.9   137.5
C00028        000    741025  0800   3  4  2   2               69.4    56.8
C00029        000    741025  0900   1  4  2   2              464.6   393.8
C00030        000    741025  0900   2  4  2   2              141.2   150.4
C00031        000    741025  0900   3  4  2   2               36.1    62.2
C00032        000    741024  1000   1  4  2   2              411.3   393.8
C00033        000    741024  1000   2  4  2   2              164.2   150.4
C00034        000    741024  1000   3  4  2   2               31.8    62.2
C00035        000    741025  1100   1  4  3   2              467.2   417.7
C00036        000    741025  1100   2  4  3   2              147.4   157.4
C00037        000    741025  1100   3  4  3   2               43.7    64.1
C00038        000    741025  1200   1  4  3   2              359.0   417.7
C00039        000    741025  1200   2  4  3   2              156.6   157.4
C00040        000    741025  1200   3  4  3   2               73.3    64.1
C00041        000    741025  1300   1  4  3   2              498.8   417.7
C00042        000    741025  1300   2  4  3   2              186.4   157.4
C00043        000    741025  1300   3  4  3   2               71.6    64.1
C00044        000    741025  1400   1  4  3   2              442.0   417.7
C00045        000    741025  1400   2  4  3   2              195.5   157.4
C00046        000    741025  1400   3  4  3   2               40.8    64.1
C00047        000    741025  1500   1  4  3   2              313.1   417.7
C00048        000    741025  1500   2  4  3   2              140.8   157.4
C00049        000    741025  1500   3  4  3   2               48.9    64.1
C00050        000    741031  1700   1  5  1   2              194.7   277.3
C00051        000    741031  1700   2  5  1   2               40.0   130.8
C00052        000    741031  1800   1  5  1   3              206.8   491.1
C00053        000    741031  1800   2  5  1   3              118.2   224.5
C00054        000    741031  1800   3  5  1   3               25.9   103.0
C00055        000    741031  1900   1  4  1   3              119.0   145.5
```

```
C00056     000   741031   1900   2  4  1   3           73.1    73.4
C00057     000   741031   1900   3  4  1   3           22.6    41.3
C00058     000   741031   2000   1  5  1   3          201.1   234.1
C00059     000   741031   2000   2  5  1   3           77.4   111.7
C00060     000   741031   2000   3  5  1   3           38.6    61.3
C00061     000   741031   2100   1  5  1   3          137.8   292.5
C00062     000   741031   2100   2  5  1   3          137.1   137.5
C00063     000   741031   2100   3  5  1   3           34.3    71.0
C00064     000   741104   0700   3  4  2   3           16.6    53.7
C00065     000   741114   0300   1  4  2   2          331.3   291.0
C00066     000   741114   0300   2  4  2   2          153.4   119.4
C00067     000   741114   0300   3  4  2   2           58.5    53.9
C00068     000   741114   0400   2  4  2   2          140.3   141.4
C00069     000   741114   0400   3  4  2   2           61.4    59.9
C00070     000   741114   0500   1  4  3   2          597.6   417.7
C00071     000   741114   0500   2  4  3   2          170.4   157.4
C00072     000   741114   0500   3  4  3   2           52.1    64.1
C00073     000   741114   0600   1  4  3   2          434.2   417.7
C00074     000   741114   0600   2  4  3   2          157.3   157.4
C00075     000   741114   0600   3  4  3   2           54.6    64.1
C00076     000   741114   0700   1  4  3   2          467.0   417.7
C00077     000   741114   0700   2  4  3   2          146.4   157.7
C00078     000   741114   0700   3  4  3   2           53.3    64.1
C00079     000   741114   0800   1  4  3   2          470.0   417.7
C00080     000   741114   0800   2  4  3   2          163.8   157.7
C00081     000   741114   0800   3  4  3   2           40.8    64.1
C00082     000   741114   0900   1  4  3   2          519.8   417.7
C00083     000   741114   0900   2  4  3   2          134.2   157.7
C00084     000   741114   0900   3  4  3   2           55.7    64.1
C00085     000   741114   1000   1  4  3   2          462.7   417.7
C00086     000   741114   1000   2  4  3   2          170.0   157.7
C00087     000   741114   1000   3  4  3   2           54.8    64.1
C00088     000   741114   1100   1  4  3   2          454.0   417.7
C00089     000   741114   1100   2  4  3   2          161.6   157.7
C00090     000   741114   1100   3  4  3   2           57.7    64.1
C00091     000   741114   1200   1  4  3   2          538.6   417.7
C00092     000   741114   1200   2  4  3   2          160.9   157.7
C00093     000   741114   1200   3  4  3   2           54.3    64.1
C00094     000   741120   0200   1  4  1   1           72.8    73.0
C00095     000   741120   0200   2  4  1   1           74.6    40.8
C00096     000   741120   0200   3  4  1   1           28.6    31.9
C00097     000   741120   0300   1  4  1   1           65.0    69.5
C00098     000   741120   0300   2  4  1   1           49.5    38.7
C00099     000   741120   0300   3  4  1   1           12.4    30.9
C00100     000   741120   0400   1  4  1   1           37.0    68.3
C00101     000   741120   0400   2  4  1   1           23.0    37.9

END ELT.
```

APPENDIX B

MODIFIED EPA FREQUENCY DISTRIBUTION SOFTWARE

B.1 Run Stream Example, SRI*SRI.DATAFREQ

```
@ELT,L S.DATAFREQ
ELT007 SL73R1 11/01/78 09:35:28 (7,)
000001        007   @XQT SRI*SRI.FREQ
000002        007   101 2 1 3 10 8 -3 3
000003        007   @ADD SRI*SRI.COVS2R
000004        007   W
000005        007   -1,-1
000006        007   11.5,10
000007        007   C
000008        003   @XQT SRI*SRI.FREQ
000009        007   101 2 1 3 10 8 1 1
000010        003   @ADD SRI*SRI.COVS2R
000011        003   W
000012        003   -1,-1
000013        007   11.5,-1,10
000014        003   C
000015        003   @XQT SRI*SRI.FREQ
000016        007   101 2 1 4 10 8 .1 2
000017        003   @ADD SRI*SRI.COVS2R
000018        003   W
000019        003   -1,-1
000020        007   11.5,10
000021        003   C
000022        003   @

END ELT.
```

Reproduced from best available copy.

B.2 Map of Program, SRI*SRI.MAPFREQ

```
@ELT,L S.MAPFREQ
ELT007 SL73R1 11/01/78 09:36:46 (4,)
C00001       000   @FOR SRI*SRI.FREQ
C00002       000   @MAP ,SRI*SRI.FREQ
C00003       000   IN SRI*SRI.FREQ
C00004       000   IN SRI*SRI.ORDER
C00005       000   IN SRI*SRI.GRALIN
C00006       000   IN SRI*SRI.GRAPH
C00007       000   IN SRI*SRI.PROB
C00008       004   IN SRI*SRI.HEADER
C00009       004   IN SRI*SRI.BOX
C00010       004   IN RAPS*UTILITY.COMPOZ
C00011       000   @ADD GRAPH*TEKTRONIX.PREVIEW/MAPECL

END ELT.
```

B.3 Listing of the Main Program, SRI*SRI.FREQ

```
@ELT,L S.FREQ
ELT007 SL73R1 11/01/78 09:35:38 (53,)
000001        049   C PROGRAM SRI*SRI.FREQ
000002        049   C MODIFIED FROM M21ADD*GMHWY.FREQ
000003        049   C PROGRAM FROM BILL PETERSON
000004        049   C MODIFIED BY ROD ALLEN, COMP-AID INC., OCTOBER 1978
000005        049   C TO COMPILE AND MAP @ADD SRI*SRI.MAPFREQ
000006        049   C TO EXECUTE AND DATA @ADD,L SRI*SRI.DATAFREQ
000007        049   C DECLARATIONS
000008        049         DIMENSION OB(999),PR(999),PMO(999),PMODO(999),OPMODO(999),
000009        049        1OPMO(999),OOB(999),FREQ(999),X(999),Y(999),YY(999),OPR(999)
000010        049   C INITIALIZATION
000011        049         WRITE(6,800)
000012        049   800   FORMAT(' ENTER LINE OF CONTROL PARAMETERS')
000013        049         READ(5,500) N,IOP,IGRAPH,NLOG,XDIM,YDIM,YBASE,NPLT
000014        049   500   FORMAT( )
000015        049   C NLOG=NUMBER OF LOG CYCLES.
000016        049   C XDIM=X DIMENSION IN INCHES OF PLOT.
000017        049   C YDIM=Y DIMENSION IN INCHES OF PLOT.FOR NPLT=3, YDIM=10.
000018        049   C YBASE=LOWEST Y VALUE.FOR NPLT=3,YBASE=-3 ONLY.
000019        049   C NPLT, NPLT=1 PLOT OBSERVED,PREDICTED VS. CUM. FREQ.
000020        049   C NPLT=2 PLOT ABS(PR-OB) VS.CUM.FREQ.
000021        049   C NPLT=3 PLOT (PR-OB)/OB VS. CUM.FREQ.
000022        049   C IOP =1,PRINTOUT OF TABLE. IOP=2,NO PRINTOUT.
000023        049   C IGRAPH=1,GRAPH DISPLAY. IGRAPH=2,NO GRAPH DISPLAY.
000024        049   C N= NUMBER OF DATA POINTS.
000025        049   C COMPUTATION
000026        049         WRITE(6,801)
000027        049   801   FORMAT(' ENTER OBSERVED AND PREDICTED DATA VALUES')
000028        049         DO 1 I=1,N
000029        049         READ(5,100) OB(I),PR(I)
000030        049   100   FORMAT(33X,2F8.0)
000031        049   C CHECK OBSERVED AND PREDICTED VALUES BECAUSE OF LOGS
000032        049         IF(OB(I).LT.0.001) OB(I)=0.001
000033        049         IF(PR(I).LT.0.001) PR(I)=0.001
000034        049         PMO(I)=PR(I)-OB(I)
000035        049   1     PMODO(I)=PMO(I)/OB(I)
000036        049         J=MOD(N,2)
000037        049         IF(J.EQ.1) GO TO 25
000038        049         NN=N/2
000039        049         J=N+1
000040        049         DO 15 I=1,NN
000041        049         FREQ(I)=100.*((I-0.4)/N)
000042        049         J=J-1
000043        049         FREQ(J)=100.-FREQ(I)
000044        049   15    CONTINUE
000045        049         GO TO 45
000046        049   25    CONTINUE
000047        049         NN=(N-1)/2
000048        049         J=N+1
000049        049         DO 35 I=1,NN
000050        049         FREQ(I)=100.*((I-0.4)/N)
000051        049         J=J-1
000052        049         FREQ(J)=100.-FREQ(I)
000053        049   35    CONTINUE
000054        049         J=NN+1
000055        049         FREQ(J)=100.*((J-0.4)/N)
```

```
049    45     CONTINUE
049             DO 47 I=1,N
049             FREQ(I)=FREQ(I)/100.
049    47       CONTINUE
049           CALL ORDER(OB,OOB,N)
049           CALL ORDER(PR,OPR,N)
049           CALL ORDER(PMO,OPMO,N)
049           CALL ORDER(PMODO,OPMODO,N)
049           IF(IOP.EQ.2) GO TO 56
049           WRITE(6,200)
049    200    FORMAT(1X,'NUM',T11,'OBS',T23,'PR',T36,'PR-OBS',T44,'(PR-OBS)/OBS'
049          1,T62,'OBS',T76,'PR',T88,'PR-OBS',T96,'(PR-OBS)/OBS',T114,'FREQ',/)
049           DO 55 I=1,N
049           WRITE(6,300) I,OB(I),PR(I),PMO(I),PMODO(I),OOB(I),OPR(I),
049          1OPMO(I),OPMODO(I),FREQ(I)
049    300    FORMAT(1X,I3,T5,9(F10.3,3X))
049    55     CONTINUE
049    56       CONTINUE
049           IF(IGRAPH.EQ.2) GO TO 4
049           CALL PLOTS(BUF,1,14)
049           CALL HEADER
051           CALL SYMBOL(2.,-.8,.14,'CUMULATIVE FREQUENCY (%)',0.,24)
049           CALL BOX(0,0,1023,780)
049           IF(NPLT.EQ.3) GO TO 95
049            CALL GRAPH(XDIM,YDIM,NLOG,YBASE,1)
049           IF(NPLT.EQ.2) GO TO 75
049    C NPLT=1
051           CALL SYMBOL(-.5,1.5,.14,'CONCENTRATION',90.,13)
051           YZ=YDIM+.5
051           CALL SYMBOL(-.5,YZ ,.14,'FREQUENCY DISTRIBUTION FOR OBSERVED AND P
049          1REDICTED CONCENTRATION',0.,63)
049            DO 65 I=1,N
049            A=FREQ(I)
049            B=OOB(I)
049            C=OPR(I)
049            X(I)=PROB(A)
049            Y(I)=ALOG10(B)
049            YY(I)=ALOG10(C)
049    65      CONTINUE
049            X(N+1)=PROB(0.001)
049            X(N+2)=(PROB(0.999)-PROB(0.001))/XDIM
049            Y(N+1)=ALOG10(YBASE)
049            YY(N+1)=Y(N+1)
049            Y(N+2)=NLOG/YDIM
049            YY(N+2)=Y(N+2)
049            CALL PLOT(0.,0.,-3)
049            CALL LINE(X,Y,N,1,-1,0)
049            CALL LINE(X,YY,N,1,-1,2)
049           CALL SYMBOL(XDIM-2.,1.,.2,0,0.,-1)
052           CALL SYMBOL(XDIM-1.7,1.,.14,'OBSERVED',0.,8)
049           CALL SYMBOL(XDIM-2.,.5,.2,2,0.,-1)
052           CALL SYMBOL(XDIM-1.7,.5,.14,'PREDICTED',0.,9)
049           GO TO 999
049    75     CONTINUE
049    C NPLT=2
052           CALL SYMBOL(-.5,0.,.14,'ABSOLUTE VALUE OF RESIDUAL CONCENTRATION',
049          1 90.,40)
```

```
052           YZ=YDIM+.5
053           CALL SYMBOL(-.6,YZ,.14,'FREQUENCY DISTRIBUTION FOR RESIDUAL (OBSER
053          1VED-PREDICTED) CONCENTRATION',0.,70)
049           IN=1
049           DO 85 I=1,N
049            A=FREQ(I)
049           IF(OPMO(I).LE.0.) IN=I+1
049            B=ABS(OPMO(I))
049           IF(B.LE.0.) B=0.001
049            X(I)=PROB(A)
049            Y(I)=ALOG10(B)
049     85     CONTINUE
049            X(N+1)=PROB(0.001)
049            X(N+2)=(PROB(0.999)-PROB(0.001))/XDIM
049            Y(N+1)=ALOG10(YBASE)
049            Y(N+2)=NLOG/YDIM
049            CALL PLOT(0.,0.,-3)
052            CALL LINE(X,Y,N,1,-1,4)
049           NIN=N-IN+1
052           CALL LINE(X(IN),Y(IN),NIN,1,-1,0)
052           CALL SYMBOL(XDIM-2.5,1.,.2,4,0.,-1)
052           CALL SYMBOL(XDIM-2.2,1.,.14,'UNDERESTIMATED',0.,14)
052           CALL SYMBOL(XDIM-2.5,.5,.2,4,0.,-1)
052           CALL SYMBOL(XDIM-2.5,.5,.2,0,0.,-1)
052           CALL SYMBOL(XDIM-2.2,.5,.14,'OVERESTIMATED',0.,13)
049           GO TO 999
049     95    CONTINUE
049     C NPLT=3
052           CALL SYMBOL(-.5,2.,.14,'RESIDUAL (%)',90.,12)
052           YZ=YDIM+.5
053           CALL SYMBOL(0.,YZ,.14,'FREQUENCY DISTRIBUTION FOR PERCENTAGE',0.,
049          1 37)
053           YZ=YDIM+.25
053           CALL SYMBOL(0.,YZ ,.14,'OF RESIDUAL RELATIVE TO OBSERVED CONCENTRA
049          1TION',0.,46)
049           CALL GRALIN(XDIM,YDIM,NLOG,YBASE,1)
049           DO 105 I=1,N
049           A=FREQ(I)
049           X(I)=PROB(A)
049           Y(I)=OPMODO(I)
049     105   CONTINUE
049            X(N+1)=PROB(0.001)
049            X(N+2)=(PROB(0.999)-PROB(0.001))/XDIM
049           Y(N+1)=YBASE
049           Y(N+2)=10./YDIM
049           CALL PLOT(0.,0.,-3)
049           CALL LINE(X,Y,N,1,-1,0)
049     C TERMINATION
049     999   CALL PLOT(0.,0.,999)
049           CALL HDCOPY
049     4     STOP
049           END

END ELT.
```

B.4 Subroutine SRI*SRI.GRALIN

```
@ELT,L S.GRALIN
ELT007 SL73R1 11/01/78 09:36:00 (12,)
008             SUBROUTINE GRALIN(XDIM,YDIM,NLOG,YBASE,NPLOT)                   GRAPH010
012    C MODIFIED BY ROD ALLEN, COMP-AID INC., OCTOBER 1978
008    C        GRAPH DRAWS THE BASE CHART WITH PROBABILITY SCALE ON THE       GRAPH020
011    C        ABSCISSA AND LINEAR SCALE ON THE ORDINATE. THIS ROUTINE WAS       GRAPH030
008    C        DEVELOPED BY DALE COVENTRY.                                    GRAPH040
008          DIMENSION XVAL(21)                                                GRAPH050
008          DATA XVAL/0.1,0.2,0.5,1.0,2.0,5.0,10.0,20.0,30.0,40.0,50.0,60.0,  GRAPH060
008         U 70.0,80.0,90.0,95.0,98.0,99.0,99.5,99.8,99.9/                    GRAPH070
008           DATA DUMT/' '/
008          IF(NPLOT.GT.1)GO TO 17                                            GRAPH110
008    C        SET TO BASE PLOTTER POSITION AT BOTTOM OF PAGE.                GRAPH120
012    CR    CALL PLOT(12.0,-36.0,3)                                           GRAPH130
012    CR    CALL PLOT((XDIM/2.),-33.0,-3)                                     GRAPH140
012          CALL PLOT((XDIM/2.),0.,-3)
008          FCTR=ABS((XDIM/2.)/PROB(0.001))                                   GRAPH150
008          GO TO 18                                                          GRAPH160
008    17    ICHECK=MOD(NPLOT,3)                                               GRAPH170
008    C        GO TO CORRECT POSITION FOR MODULUS 3 OF PLOT NUMBER            GRAPH180
008    C        (THREE PLOTS ARE MADE ACROSS WIDTH OF PAPER.)                  GRAPH190
012    CR    IF(ICHECK.EQ.0)CALL PLOT(5.,11.,-3)                               GRAPH200
012    CR    IF(ICHECK.EG.1)CALL PLOT(17.,-22.,-3)                             GRAPH210
012    CR    IF(ICHECK.EQ.2)CALL PLOT(5.,11.,-3)                               GRAPH220
012    18    M=-1                                                              GRAPH230
008    C        INDICATE ON PRINTER THAT PLOT IS BEING ATTEMPTED.              GRAPH240
008    C        POSITION AT CENTER OF PLOT                                     GRAPH260
008          CALL SYMBOL((-XDIM/2.),0.0,0.1,13,0.0,-1)                         GRAPH270
008    C        DRAW CENTER LINE                                               GRAPH280
008          CALL PLOT((-XDIM/2.),0.0,3)                                       GRAPH290
008          CALL PLOT((-XDIM/2.),YDIM,2)                                      GRAPH300
008          CALL SYMBOL((-XDIM/2.),YDIM,0.1,13,0.0,-1)                        GRAPH310
008    C        LOOP TO DRAW LINES AND PLOT TOP AND BOTTOM NUMBERS FOR         GRAPH320
008    C        PROBABILITY PART OF GRAPH.                                     GRAPH330
008          DO 2 I=2,21                                                       GRAPH340
008          M=M*(-1)                                                          GRAPH350
008    C        M IS A 'FLIP-FLOP' SWITCH TO GO FROM TOP TO BOTTOM, THEN BOTTOMGRAPH360
008    C        TO TOP, ETC.                                                   GRAPH370
008          ANUM=XVAL(I)                                                      GRAPH380
008          BNUM=100.0-XVAL(I)                                                GRAPH390
008          CNUM=XVAL(I)*0.01                                                 GRAPH400
008    C        DETERMINE LINE POSITION                                        GRAPH410
008          XP=PROB(CNUM)*FCTR                                                GRAPH420
008          IF(M.LT.0)GO TO 1                                                 GRAPH430
008          IF(ANUM.LT.10.)GO TO 11                                           GRAPH440
008          CALL NUMBER((XP-0.17),-0.15,0.07,ANUM,0.0,1)                      GRAPH450
008          GO TO 12                                                          GRAPH460
008    11    CALL NUMBER((XP-0.07),-0.15,0.07,ANUM,0.0,1)                      GRAPH470
008    12    CALL SYMBOL(XP,0.0,0.1,13,0.0,-1)                                 GRAPH480
008          CALL PLOT(XF,0.0,3)                                               GRAPH490
008          CALL PLOT(XP,YDIM,2)                                              GRAPH500
008          CALL SYMBOL(XP,YDIM,0.1,13,0.0,-1)                                GRAPH510
008          IF(BNUM.LT.10.)GO TO 13                                           GRAPH520
008          CALL NUMBER((XP-0.17),(YDIM+0.05),0.07,BNUM,0.0,1)                GRAPH530
008          GO TO 2                                                           GRAPH540
008    13    CALL NUMBER((XP-0.07),(YDIM+0.05),0.07,BNUM,0.0,1)                GRAPH550
008          GO TO 2                                                           GRAPH560
```

```
000056      008   1      IF(BNUM.LT.10.)GO TO 14                                                   GRAPH570
000057      008          CALL NUMBER((XP-0.17),(YDIM+0.05),0.07,BNUM,0.0,1)                        GRAPH580
000058      008          GO TO 15                                                                  GRAPH590
000059      008   14     CALL NUMBER((XP-0.07),(YDIM+0.05),0.07,BNUM,0.0,1)                        GRAPH600
000060      008   15     CALL SYMBOL(XP,YDIM,0.1,13,0.0,-1)                                        GRAPH610
000061      008          CALL PLOT(XP,YDIM,3)                                                      GRAPH620
000062      008          CALL PLOT(XP,0.0,2)                                                       GRAPH630
000063      008          CALL SYMBOL(XP,0.0,0.1,13,0.0,-1)                                         GRAPH640
000064      008          IF(ANUM.LT.10.)GO TO 16                                                   GRAPH650
000065      008          CALL NUMBER((XP-0.17),-0.15,0.07,ANUM,0.0,1)                              GRAPH660
000066      008          GO TO 2                                                                   GRAPH670
000067      008   16     CALL NUMBER((XP-0.07),-0.15,0.07,ANUM,0.0,1)                              GRAPH680
000068      008   2      CONTINUE                                                                  GRAPH690
000069      008   C         WRITE ORDINATE LEGEND.(TEXT AND NUMBERS)                               GRAPH700
000070      008          XCOR=XDIM/(-2.)
000071      008          CALL AXIS(XCOR,0.0,DUMT,1,YDIM,90.,YBASE,(10./YDIM))
000072      008           CALL PLOT(XCOR,0.0,3)
000073      008           DO 50 I=1,11
000074      010           CALL PLOT((XCOR+XDIM),(I-1.),2)
000075      010           CALL PLOT(XCOR,(I-0.),3)
000076      008   50      CONTINUE
000077      008   31     CALL PLOT(XCOR,0.0,-3)                                                     GRAPH960
000078      008          RETURN                                                                    GRAPH970
000079      008          END                                                                       GRAPH980

END ELT.
```

B.4 Subroutine SRI*SRI.GRAPH

```
@ELT,L S.GRAPH
ELT007 SL73R1 11/01/78 09:36:11 (6,)
000001       002            SUBROUTINE GRAPH(XDIM,YDIM,NLOG,YBASE,NPLOT)                      GRAPH010
000002       006      C MODIFIED BY ROD ALLEN, COMP-AID INC., OCTOBER 1978
000003       002      C        GRAPH DRAWS THE BASE CHART WITH PROBABILITY SCALE ON THE       GRAPH020
000004       002      C        ABSCISSA AND LOG SCALE ON THE ORDINATE. THIS ROUTINE WAS       GRAPH030
000005       002      C        DEVELOPED BY DALE COVENTRY.                                    GRAPH040
000006       003            DIMENSION XVAL(21)                                                GRAPH050
000007       002            DATA XVAL/0.1,0.2,0.5,1.0,2.0,5.0,10.0,20.0,30.0,40.0,50.0,60.0,  GRAPH060
000008       002           U 70.0,80.0,90.0,95.0,98.0,99.0,99.5,99.8,99.9/                    GRAPH070
000009       005             DATA DUMT/' '/
000010       002            IF(NPLOT.GT.1)GO TO 17                                            GRAPH110
000011       002      C        SET TO BASE PLOTTER POSITION AT BOTTOM OF PAGE.                GRAPH120
000012       006      CR    CALL PLOT(12.0,-36.0,3)                                           GRAPH130
000013       006      CR    CALL PLOT((XDIM/2.),-33.0,-3)                                     GRAPH140
000014       006            CALL PLOT((XDIM/2.),0.,-3)
000015       002            FCTR=ABS((XDIM/2.)/PROB(0.001))                                   GRAPH150
000016       002            GO TO 18                                                          GRAPH160
000017       002      17    ICHECK=MOD(NPLOT,3)                                               GRAPH170
000018       002      C        GO TO CORRECT POSITION FOR MODULUS 3 OF PLOT NUMBER            GRAPH180
000019       002      C        (THREE PLOTS ARE MADE ACROSS WIDTH OF PAPER.)                  GRAPH190
000020       006      CR    IF(ICHECK.EQ.0)CALL PLOT(5.,11.,-3)                               GRAPH200
000021       006      CR    IF(ICHECK.EQ.1)CALL PLOT(17.,-22.,-3)                             GRAPH210
000022       006      CR    IF(ICHECK.EQ.2)CALL PLOT(5.,11.,-3)                               GRAPH220
000023       002      18    M=-1                                                              GRAPH230
000024       002      C        INDICATE ON PRINTER THAT PLOT IS BEING ATTEMPTED.              GRAPH240
000025       002      C        POSITION AT CENTER OF PLOT                                     GRAPH260
000026       002            CALL SYMBOL((-XDIM/2.),0.0,0.1,13,0.0,-1)                         GRAPH270
000027       002      C        DRAW CENTER LINE                                               GRAPH280
000028       002            CALL PLOT((-XDIM/2.),0.0,3)                                       GRAPH290
000029       002            CALL PLOT((-XDIM/2.),YDIM,2)                                      GRAPH300
000030       002            CALL SYMBOL((-XDIM/2.),YDIM,0.1,13,0.0,-1)                        GRAPH310
000031       002      C        LOOP TO DRAW LINES AND PLOT TOP AND BOTTOM NUMBERS FOR         GRAPH320
000032       002      C        PROBABILITY PART OF GRAPH.                                     GRAPH330
000033       002            DO 2 I=2,21                                                       GRAPH340
000034       002            M=M*(-1)                                                          GRAPH350
000035       002      C        M IS A 'FLIP-FLOP' SWITCH TO GO FROM TOP TO BOTTOM, THEN BOTTOMGRAPH360
000036       002      C        TO TOP, ETC.                                                   GRAPH370
000037       002            ANUM=XVAL(I)                                                      GRAPH380
000038       002            BNUM=100.0-XVAL(I)                                                GRAPH390
000039       002            CNUM=XVAL(I)*0.01                                                 GRAPH400
000040       002      C        DETERMINE LINE POSITION                                        GRAPH410
000041       002            XP=PROB(CNUM)*FCTR                                                GRAPH420
000042       002            IF(M.LT.0)GO TO 1                                                 GRAPH430
000043       002            IF(ANUM.LT.10.)GO TO 11                                           GRAPH440
000044       002            CALL NUMBER((XP-0.17),-0.15,0.07,ANUM,0.0,1)                      GRAPH450
000045       002            GO TO 12                                                          GRAPH460
000046       002      11    CALL NUMBER((XP-0.07),-0.15,0.07,ANUM,0.0,1)                      GRAPH470
000047       002      12    CALL SYMBOL(XP,0.0,0.1,13,0.0,-1)                                 GRAPH480
000048       002            CALL PLOT(XP,0.0,3)                                               GRAPH490
000049       002            CALL PLOT(XP,YDIM,2)                                              GRAPH500
000050       002            CALL SYMBOL(XP,YDIM,0.1,13,0.0,-1)                                GRAPH510
000051       002            IF(BNUM.LT.10.)GO TO 13                                           GRAPH520
000052       002            CALL NUMBER((XP-0.17),(YDIM+0.05),0.07,BNUM,0.0,1)                GRAPH530
000053       002            GO TO 2                                                           GRAPH540
000054       002      13    CALL NUMBER((XP-0.07),(YDIM+0.05),0.07,BNUM,0.0,1)                GRAPH550
000055       002            GO TO 2                                                           GRAPH560
```

```
C00056     002   1     IF(BNUM.LT.10.)GO TO 14                                              GRAPH57
C00057     002         CALL NUMBER((XP-0.17),(YDIM+0.05),0.07,BNUM,0.0,1)                    GRAPH58
C00058     002         GO TO 15                                                             GRAPH59
C00059     002   14    CALL NUMBER((XP-0.07),(YDIM+0.05),0.07,BNUM,0.0,1)                    GRAPH60
C00060     002   15    CALL SYMBOL(XP,YDIM,0.1,13,0.0,-1)                                    GRAPH61
C00061     002         CALL PLOT(XP,YDIM,3)                                                  GRAPH62
C00062     002         CALL PLOT(XP,0.0,2)                                                   GRAPH63
C00063     002         CALL SYMBOL(XP,0.0,0.1,13,0.0,-1)                                     GRAPH64
C00064     002         IF(ANUM.LT.10.)GO TO 16                                              GRAPH65
C00065     002         CALL NUMBER((XP-0.17),-0.15,0.07,ANUM,0.0,1)                          GRAPH66
C00066     002         GO TO 2                                                              GRAPH67
C00067     002   16    CALL NUMBER((XP-0.07),-0.15,0.07,ANUM,0.0,1)                          GRAPH68
C00068     002   2     CONTINUE                                                             GRAPH69
C00069     002   C        WRITE ORDINATE LEGEND.(TEXT AND NUMBERS)                          GRAPH70
C00070     005         CALL LGAXS((XDIM/(-2.)),0.0,DUMT,1,YDIM,90.,YBASE,(NLOG/YDIM))        GRAPH7
C00071     002         XPLOT=XDIM/2.0                                                       GRAPH72
C00072     002         ZPLOT=-1.0*XPLOT                                                     GRAPH73
C00073     002         M=-1                                                                 GRAPH74
C00074     002   C        LOOP TO DRAW LINES OF LOG PART OF GRAPH                           GRAPH75
C00075     002         DO 4 I=1,NLOG                                                        GRAPH76
C00076     002         DO 4 J=1,9                                                           GRAPH77
C00077     002         AJ=J                                                                 GRAPH78
C00078     002   C        "FLIP-FLOP" TO GO FROM LEFT TO RIGHT, THEN RIGHT TO LEFT, ETC.    GRAPH79
C00079     002         M=M*(-1)                                                             GRAPH80
C00080     002         YPLOT=(I-1)*(YDIM/NLOG)+ALOG10(AJ)*(YDIM/NLOG)                       GRAPH81
C00081     002         IF(M.LT.0)GO TO 3                                                    GRAPH82
C00082     002         CALL PLOT(XPLOT,YPLOT,3)                                             GRAPH83
C00083     002         CALL PLOT(ZPLOT,YPLOT,2)                                             GRAPH84
C00084     002         GO TO 4                                                              GRAPH85
C00085     002   3     CALL PLOT(ZPLOT,YPLOT,3)                                             GRAPH86
C00086     002         CALL PLOT(XPLOT,YPLOT,2)                                             GRAPH8
C00087     002   4     CONTINUE                                                             GRAPH8
C00088     002   C        DRAW TOP LINE                                                     GRAPH8
C00089     002         IF(M.LT.0)GO TO 30                                                   GRAPH9
C00090     002         CALL PLOT(XPLOT,YDIM,3)                                              GRAPH9
C00091     002         CALL PLOT(ZPLOT,YDIM,2)                                              GRAPH9
C00092     002         GO TO 31                                                             GRAPH9
C00093     002   30    CALL PLOT(ZPLOT,YDIM,3)                                              GRAPH9
C00094     002         CALL PLOT(XPLOT,YDIM,2)                                              GRAPH9
C00095     002   31    CALL PLOT(ZPLOT,0.0,-3)                                              GRAPH9
C00096     002         RETURN                                                               GRAPH9
C00097     002         END                                                                  GRAPH9

END ELT.
```

B.4 Subroutine SRI*SRI.ORDER

```
@ELT,L S.ORDER
ELT007 SL73R1 11/01/78 09:37:01 (2,)
000001       000            SUBROUTINE ORDER(A,B,N)
000002       001      C SUBROUTINE ORDER FROM BILL PETERSON
000003       001      C USED BY PROGRAM FREQ
000004       002            DIMENSION A(1),B(1)
000005       000            DO 10 I=1,N
000006       000    10      B(I)=A(I)
000007       000            NN=N-1
000008       000            DO 20 K=1,NN
000009       000            DO 30 J=1,NN
000010       000            IF(B(J)-B(J+1)) 30,25,25
000011       000    25      X=B(J)
000012       000            B(J)=B(J+1)
000013       000            B(J+1)=X
000014       000    30      CONTINUE
000015       000    20      CONTINUE
000016       000            RETURN
000017       000            END

END ELT.
```

B.4 Subroutine SRI*SRI.PROB

```
@ELT,L S.PROB
ELT007 SL73R1 11/01/78 09:37:08 (3,)
C00001        000          FUNCTION PROB(Z)                                                    PROB0010
C00002        000    C***THE FUNCTION PROB(Z) TAKES THE INPUT Z IN FREQUENCY                  PROB0020
C00003        000    C***FROM 0 TO 1.0 AND DETERMINES THE PLUS OR MINUS                       PROB0030
C00004        000    C***STANDARD DEVIATIONS(PROB) ON THE NORMAL CURVE OF                     PROB0040
C00005        000    C***FREQUENCY THUS GIVING A PLOTTING POSITION FOR GRAPHING               PROB0050
C00006        000    C***ON THE PROBABILITY SCALE.                                            PROB0060
C00007        000    C***THIS SUBROUTINE WAS DEVELOPED BY JOE SANTNER AND OBTAINED            PROB0070
C00008        000    C***FROM RALPH LARSEN.                                                   PROB0080
C00009        000          W(A)=SQRT(ALOG(1./(A*A)))                                           PROB0090
C00010        000          X(A)=W(A)-((2.515517+0.802853*W(A)+0.010328*(W(A)*W(A)))/(1.+      PROB0100
C00011        000         1 1.432788*W(A)+0.189269*(W(A)*W(A))+0.001308*(W(A)*W(A)*W(A))))   PROB0110
C00012        003    C      WRITE(6,900) Z
C00013        002    900    FORMAT(3X,'Z= ',E10.5)
C00014        000          IF(Z-0.5)1,7,8                                                      PROB0120
C00015        000    1     IF(Z-0.0)2,4,6                                                      PROB0130
C00016        000    2     WRITE(6,3)Z                                                         PROB0140
C00017        000    3     FORMAT(2X,'INPUT ERROR PROB =',E20.10)                              PROB0150
C00018        000          GO TO 10                                                            PROB0160
C00019        000    4     WRITE(6,5)Z                                                         PROB0170
C00020        000    5     FORMAT(2X,'ANS DEVIA EQUAL MINUS OR PLUS INFINITY PROB =',E17.10) PROB0180
C00021        000          GO TO 10                                                            PROB0190
C00022        000    6     PROB=-X(Z)                                                          PROB0200
C00023        000          GO TO 10                                                            PROB0210
C00024        000    7     PROB=0.0                                                            PROB0220
C00025        000          GO TO 10                                                            PROB0230
C00026        000    8     IF(Z-1.0)9,4,2                                                      PROB0240
C00027        000    9     PROB=X(1.0-Z)                                                       PROB0250
C00028        000    10    RETURN                                                              PROB0260
C00029        000          END                                                                 PROB0270

END ELT.
```

APPENDIX C

ACCURACY SCORE LISTINGS

C.1 Listing of the Main Program, M21ADO*STAT01.ACSCOR

```
@FLT,L S1.ACSCOR
ELT007 SL73R1 11/01/78 09:37:57 (28,)
        025   C     ****************************************************************
        025   C
        025   C           PROGRAM ACSCOR COMPUTES AN ACCURACY SCORE FOR UP TO EIGHT
        025   C     DIFFERENT LOSS FUNCTIONS.   FOR EACH SCORE THE CONFIDENCE INTER-
        025   C     VAL IS ALSO COMPUTED.  AS AN OPTION THE ACCURACY SCORE FOR EACH
        025   C     STATION CAN BE ENTERED ON A MAP OF ST. LOUIS USING THE TEKTRONIX
        025   C     GRAPHICS MODEL 4014 TERMINAL.  OPTIONS EXIST WITHIN THE PROGRAM
        025   C     TO COMPUTE THE ACCURACY SCORE FOR ALL STATIONS COMBINED, A SPECI-
        025   C     FIED STATION, AND EACH STATION.
        025   C
        025   C     ****************************************************************
        024         COMMON /VAR/IDATE(123),ITIME(123),ISITE(123)
        024         COMMON /DAT/E1,E2,E3,E4,PCMIN,HL1,HL2,CMAX,DD(5,5),AS(5,5)
        024         DIMENSION OC(123),PC(123),BS(5,5),DS(5,5),NOT(8),A(10),SOCC(123,3)
        024        1,SPCC(123,3),IDATEC(123,3),ITIMEC(123,3),ISITEC(123,3),OCA(123,5),
        024        2PCA(123,5),SOCA(123,5,3),SPCA(123,5,3),KK(3),NOS(26),NOD(10)
        024        3,NODS(10,3),LM(3),ENORM(3),ERR(25)
        024         INTEGER OUT
        024         DATA NOS /3R 01,3R 02,3R 03,3R 04,3R 05,3R 06,3R 07,3R 08,3R 09,
        024        13R 10,3R 11,3R 12,3R 13,3R 14,3R 15,3R 16,3R 17,3R 18,3R 19,3R 20,
        024        23R 21,3R 22,3R 23,3R 24,3R 25,3RALL/
        025         DATA ERR/25*-1./
        024         DATA BS /0.0,1.0,2.0,3.0,4.0,1.0,0.0,1.0,2.0,3.0,2.0,1.0,0.0,1.0,
        024        12.0,3.0,2.0,1.0,0.0,1.0,4.0,3.0,2.0,1.0,0.0/
        024         DATA  DS /0.0,0.45,1.15,0.0,0.0,0.45,0.0,0.07,0.0,0.0,1.15,0.7,0.0
        024        1,12*0.0/
        026         DATA M,N,INP,OUT,PCMIN,NCRIT,INC /5,0,5,6,25.0,5,1/
        024     1   FORMAT (///5X56H***** STATISTICAL TECHNIQUES FOR EVALUATING MODELS
        024        1 *****)
        024     2   FORMAT (/70H # ENTER THE NO. OF DATA (ND) AND THE NO. OF MONITORIN
        024        1G STATIONS (NS)./)
        024     3   FORMAT ()
        024     4   FORMAT (/43H # ENTER ND LINES OF THE FOLLOWING DATA .../
        024        1 /6X,16H1. DATE (YRMODA)
        024        2 /6X,14H2. TIME (HRMN)
        024        3 /6X,22H3. MONITORING SITE NO.
        024        4 /6X,17H4. OBSERVED VALUE
        024        5 /6X,18H5. PREDICTED VALUE/)
        024     5   FORMAT (/45H # SELECT ONE MODE NO. FROM THE FOLLOWING ...,/
        024        1 /5X42H 1.  DETAILED-RUN (FOR UN-INITIATED USERS)
        025        2 /5X36H 2.  QUICK-RUN (FOR INITIATED USERS))
        024     7   FORMAT (/27H YOU BLEW IT ... TRY AGAIN. )
        024     8   FORMAT (/54H # SELECT TEST NOS. WITH A SERIES OF 1(YES) AND 0(NO).
        024        1 //5X25H 1.  MEAN ABSOLUTE ERROR.
        024        2  /5X23H 2.  MEAN SQUARE ERROR.
        024        3  /5X30H 3.  ABSOLUTE ERROR THRESHOLD.
        024        4  /5X32H 4.  PERCENTAGE ERROR THRESHOLD.
        024        5  /5X38H 5.  SYMMETRIC HIGH-LOW LOSS FUNCTION.
        024        6  /5X39H 6.  ASYMMETRIC HIGH-LOW LOSS FUNCTION.
        024        7  /5X31H 7.  USER SUPPLIED LOSS MATRIX.
        024        8  /5X36H 8.  MAXIMUP CONCENTRATION LOCATION./)
        024     9   FORMAT (/45H # SELECT ONE TEST NO. FROM THE FOLLOWING ...,/
        024        1  /5X28H 1.  MEAN ABSOLUTE ERROR ...
        024        1      /10X35HE = (1.0/N)*(SUM(ABS(OC(I)-PC(I)))),5X11HWHERE I=1,N
        024        2  /5X26H 2.  MEAN SQUARE ERROR ...
024        2      /10X35HE = (1.0/N)*(SUM((OC(I)/PC(I))**2)),5X11HWHERE I=1,N
024        3   /5X33H 3.  ABSOLUTE ERROR THRESHOLD ...
024        3      /10X24HE = (1.0/N)*(SUM(C(I))))               ,16X11HWHERE I=1,N
024        3   /15X14HWHERE C(I) = 0,6X25HIF ABS(OC(I)-PC(I)).LE.E1
024        3   /15X14HWHERE C(I) = 1  ,6X25HIF ABS(OC(I)-PC(I)).GT.E1
024        4   /5X35H 4.  PERCENTAGE ERROR THRESHOLD ...
024        4      /10X23HE = (1.0/N)*(SUM(C(I)))                ,17X11HWHERE I=1,N
024        4   /15X14HWHERE C(I) = 0,6X,33HIF ABS((OC(I)-PC(I))/OC(I))).LE.E2
024        4   /15X14HWHERE C(I) = 1,6X,33HIF ABS((OC(I)-PC(I))/OC(I))).GT.E2
024        5   /5X41H 5.  SYMMETRIC HIGH-LOW LOSS FUNCTION ...
024        5      /10X23HE = (1.0/N)*(SUM(C(I)))                ,17X11HWHERE I=1,N
024        5   /15X14HWHERE C(I) = 0,6X30HIF PC(I) AND OC(I).LE.NORM, E3
024        5   /35X26HIF PC(I) AND OC(I).GT.NORM
024        5   /15X14HWHERE C(I) = 1,6X13HIF OTHERWISE. )
024   10    FORMAT (/5X,42H 6.  ASYMMETRIC HIGH-LOW LOSS FUNCTION ...
024        6      /10X23HE = (1.0/N)*(SUM(C(I)))          17X,11HWHERE I=1,N
024        6   /15X15HWHERE C(I) = L1,5X20HIF PC(I).GT.NORM, E4
024        6   /35X17HAND OC(I).LT.NORM
024        6   /15X15HWHERE C(I) = L2,5X16HIF PC(I).LT.NORM
024        6   /35X17HAND OC(I).GT.NORM
024        7   /5X34H 7.  USER SUPPLIED LOSS MATRIX ...
024        7      /10X23HE = (1.0/N)*(SUM(C(I)))          17X,11HWHERE I=1,N
024        7   /15X19HWHERE C(I) = L(J,K)
024        7   /15X25HWHERE PC(I)=J AND OC(I)=K
024        8   /5X,39H 8.  MAXIMUM CONCENTRATION LOCATION ...
024        8      /10X23HE = (1.0/N)*(SUM(D(I)))          17X,11HWHERE I=1,N
024        8   /50X19HWHERE D(I)=DISTANCE/)
024   11    FORMAT (/3X5HSITE ,R3,1H,30H TEST1: MEAN ABSOLUTE ERROR IS,
024        1E10.3)
024   12    FORMAT (/57H # WOULD YOU LIKE TO TRY ANOTHER TEST ... TYPE YES OR
024        1NO. )
024   13    FORMAT (/3X5HSITE ,R3,1H,29H TEST2: MEAN SQUARED ERROR IS,E10.3)
024   14    FORMAT (/56H # SELECT A DATA SORTING CODE NO. FROM THE FOLLOWING .
024        1..//5X48H -1,   EXECUTE TEST(S) ON DATA FOR EACH STATION.
024        2   /5X48H  0,   EXECUTE TEST(S) ON DATA FOR ALL STATIONS.
024        3   /5X[illegible]H  N,   EXECUTE TEST(S) ON DATA FOR THE NTH STATION (WHERE
024        4 N=1,NS).)
024   15    FORMAT (/45H # ENTER A VALUE FOR THE ERROR THRESHOLD, E1. )
024   16    FORMAT (/3X5HSITE ,R3,1H,35H TEST3: ABSOLUTE ERROR THRESHOLD IS,
024        1E10.3)
024   17    FORMAT (/65H # DO YOU WANT TO TRY ANOTHER ERROR THRESHOLD ... TYPE
024        1 YES OR NO. )
024   18    FORMAT (A3)
024   19    FORMAT (/45H # ENTER THE PERCENTAGE ERROR THRESHOLD (E2).)
024   20    FORMAT (/3X5HSITE ,R3,1H,37H TEST4: PERCENTAGE ERROR THRESHOLD IS,
024        1 E10.3)
024   21    FORMAT (/64H # ENTER A VALUE FOR THE SYMMETRIC HI-LO LOSS FUNCTION
024        1 NORM, E3. )
024   22    FORMAT (/3X5HSITE ,R3,1H,46H TEST5: SYMMETRIC HI-LO LOSS FUNCTION
024        1ERROR IS,E10.3)
024   23    FORMAT (/89H # ENTER (1) NORM, E3, (2) UNDERPREDICTION LOSS, L1 AN
024        1D (3) THE OVERPREDICTION LOSS, L2. )
024   24    FORMAT (/3X5HSITE ,R3,1H,47H TEST6: ASYMMETRIC HI-LO LOSS FUNCTION
024        1 ERROR IS,E10.3)
024   25    FORMAT (/58H # ENTER A VALUE FOR THE MAX EXPECTED CONCENTRATION, C
024        1MAX./ )
024   26    FORMAT (/3X5HSITE ,R3,1H,42H TEST7: USER SUPPLIED LOSS MATRIX ERRO
024           1R IS E10.3)
024      27    FORMAT (/95H # ENTER A DISTANCE MATRIX (NS X NS) BY ROWS ... TYPE
024           1YES IF AVAILABLE, TYPE NO IF UNAVAILABLE. )
024      28    FORMAT (/3X5HSITE ,R3,1H,42H TEST8: MAX CONCENTRATION LOCATION LOS
024           1S IS ,E10.3)
024      29    FORMAT (/54H # DO YOU WANT TO TRY ANOTHER CMAX ... TYPE YES OR NO.
024           1)
024      30    FORMAT (/89H # ENTER A LOSS MATRIX (5X5) BY ROWS ... IF AVAILABLE
024           1TYPE YES, IF UNAVAILABLE, TYPE NO. )
024      31    FORMAT (/61H # DO YOU WANT TO TRY ANOTHER LOSS MATRIX ... TYPE YES
024           1 OR NO. )
024      32    FORMAT (/12X42H***** NO CONFIDENCE INTERVAL (N)100) *****)
024      33    FORMAT (10X5F10.3)
024      34    FORMAT (/56H # DID YOU ENTER THE MATRIX CORRECTLY ...TYPE YES OR N
024           10.)
024      35    FORKAT (10X3F10.3)
028      36    FORMAT (/57H # WOULD YOU LIKE TO TRY ANOTHER MODE ... TYPE YES OR
028           1NO.)
024      37    FORMAT (/20X26HROOT MEAN SQUARED ERROR IS,E10.3)
028      38    FORMAT (/62H # WOULD YOU LIKE TO TRY ANOTHER DATA SORT ... TYPE YE
024           1OR NO.)
025      39    FORMAT(/94H # IF YOU HAVE A TEKTRONIX 4014 AND DESIRE A SITE MAP W
025           1ITH ACCURACY SCORES ... TYPE YES OR NO. /)
025      40    FORMAT (/13H # TEST CODE?/)
028      41    FORMAT (/19H # TEST3, ENTER E1:/)
028      42    FORMAT (/19H # TEST4, ENTER E2:/)
028      43    FORMAT (/19H # TEST5, ENTER E3:/)
028      44    FORMAT (/27H # TEST6, ENTER E4, L1, L2:/)
028      45    FORMAT (/20H # TEST7, ENTER CMAX //10X,44HNOTE ... QUICK-RUN USES
028           1PROGRAM LOSS-MATRIX. /)
025      46    FORMAT (/10X,46HNOTE ... TEST8 IS VALID ONLY FOR SORT CODE -1./)
025      47    FORMAT (/50H # TEST8, (QUICK-RUN USES PROGRAM DISTANCE MATRIX)/)
025      48    FORMAT (/13H # SORT CODE?/)
025      49    FORMAT (/13H # MODE CODE?/)
024      86    FORMAT (//24H ***** END OF TEST *****/)
024    C
024    C        INPUT FROM DATA BASE FILE ...
024    C
024    C              IDATE  = DATE OF OBSERVATION
024    C              ITIME  = TIME OF OBSERVATION.
024    C              ISITE  = STATION NO.
024    C              ISTC,IWSC,IWDC ARE NOT USED.
024    C              OCS    = OBSERVED CONCENTRATION
024    C              PCS    = PREDICTED CONCENTRATION
024    C
024            CALL TESTS (123,OC,PC,E)
024            FM=M
024            ROOTM=SQRT(FM)
024            WRITE (OUT,1)
024      50    WRITE (OUT,2)
024            READ (INP,3) NC,NS
024            WRITE (OUT, 14)
025      54    PRINT 46
024      55    READ (INP,3) ICODE
024            IF (ICODE.GE.-1.AND.ICODE.LE.NS) GO TO 60
024            WRITE (OUT,7)
024            GO TO 55
```

```
024   C
024   C            ENTER ND NO. OF DATA.
024   C
024    60   LL=0
024         N=0
026         IMAP=3HNO
024         DO 65 L=1,M
024         NOD(L)=0
024    65   CONTINUE
024         IF (ICODE.EQ.-1) GO TO 80
024   C            INPUT DATA FOR PROCESSING ONLY ONE OR ALL SITES COMBINED.
024         ISN=ICODE
024         IF (ISN.EQ.0) ISN=26
024         WRITE (OUT,4)
024         DO 75 I=1,ND
024         READ (INP,3) IDATE(I),ITIME(I),ISITE(I),ISTC,IWSC,IWDC,
024        1OCS,PCS
024         IF (ISN.EQ.26) GO TO 70
024         IF (ISITE(I).NE.ISN) GO TO 75
024    70   LL=LL+1
024         IF (OCS.LT.0.0.OR.PCS.LT.0.0) GO TO 75
024         N=N+1
024         ISITE(N)=ISITE(I)
024         IDATE(N)=IDATE(I)
024         ITIME(N)=ITIME(I)
024         OC(N)=OCS
024         PC(N)=PCS
024         IF (ND.LT.100) GO TO 75
024         K=MOD(LL,M)
024         IF (K.EQ.0) K=5
024         NOD(K)=NOD(K)+1
024         LN=NOD(K)
024         OCA(LN,K)=OCS
024         PCA(LN,K)=PCS
024    75   CONTINUE
024         GO TO 95
024   C            INPUT DATA FOR PROCESSING EACH SITE.
024    80   WRITE (OUT,4)
024         DO 85 IS=1,NS
024         LM(IS)=0
024         KK(IS)=0
024         DO 85 J=1,M
024         NODS(J,IS)=0
024    85   CONTINUE
024         DO 90 I=1,ND
024         READ (INP,3) IDATE(I),ITIME(I),ISITE(I),ISTC,IWSC,IWDC,
024        1OCS,PCS
024         IS=ISITE(I)
024         LM(IS)=LM(IS)+1
024         IF (OCS.LT.0.0.OR.PCS.LT.0.0) GO TO 90
024         KK(IS)=KK(IS)+1
024         ID=KK(IS)
024         SOCC(ID,IS)=OCS
024         SPCC(ID,IS)=PCS
024         IDATEC(ID,IS)=IDATE(I)
024         ITIMEC(ID,IS)=ITIME(I)
024         ISITEC(ID,IS)=ISITE(I)
```

```
024          IF (ND.LT.100) GO TO 90
024          K=MOD(LM(IS),M)
024          IF (K.EQ.0) K=5
024          NODS(K,IS)=NODS(K,IS)+1
024          LN=NODS(K,IS)
024          SOCA(LN,K,IS)=OCS
024          SPCA(LN,K,IS)=PCS
024     90   CONTINUE
024          IF (IMC.EQ.0) GO TO 95
024          IMAP=3HYES
024          IMC=0
024          WRITE (OUT,39)
024          READ (INP,18) IYON
025          IF (IYON.NE.3HYES) IMAP=3HNO
025          IF (IYON.EQ.3HYES) CALL INITT(120)
024   C
024   C           ENTER THE DESIRED MODE.
024   C
024     95   WRITE (OUT,5)
025     99   WRITE (OUT,49)
024     100  READ (INP,3) MODE
024          IF (MODE.GE.1.AND.MODE.LE.2) GO TO 105
024          WRITE (OUT,7)
024          GO TO 100
024   C
024   C           BRANCH TO 2 DIFFERENT MODE CASES.
024   C
024     105  IF (MODE.EQ.1) GO TO 125
024   C
024   C           SHORT FORM SELECTION.
024   C
024          WRITE (OUT,8)
024     110  READ (INP,3)(NOT(L),L=1,8)
025          DO 115 I=1,8
025          IF (NOT(I).LT.0.OR.NOT(I).GT.1) GO TO 120
025     115  CONTINUE
025          GO TO 135
025     120  WRITE (OUT,7)
025          GO TO 110
024   C
024   C           LONG FORM SELECTION.
024   C
024     125  WRITE (OUT,9)
024          WRITE (OUT,10)
025          WRITE (OUT,46)
024   C
024   C           ENTER THE DESIRED TEST NUMBER.
024   C
025     130  WRITE(OUT,40)
025          READ (INP,3) NOTE
024          IF (NOTE.GE.1.AND.NOTE.LE.8) GO TO 135
024          WRITE (OUT,7)
025          GO TO 130
024   C
024   C           BRANCH TO 8 DIFFERENT TEST CASES.
024   C
025     135  IF (IMAP.EQ.3HYES) CALL HCOPY
```

```
025          GO TO  (140,150,160,170,180,190,200,225),NOTE
024    C
024    C          1.  COMPUTE THE MEAN ABSOLUTE ERROR.
024    C
024     140  IF (MODE.EQ.2.AND.NOT(1).EQ.0) GO TO 150
024          IF (ICODE.LT.0) GO TO 144
024    C          COMPUTE FOR ONE/ALL SITES.
024          CALL TEST1 (N,OC,PC,E)
024          WRITE (OUT,11) NOS(ISN),E
024          IF (LN.LT.NCRIT) GO TO 142
024    C         DETERMINE THE ERROR AND CONFIDENCE LIMITS (C.L.).
024          SUM2=0.0
024          DO 141 I=1,M
024          CALL TEST1 (NOD(I),OCA(1,I),PCA(1,I),A(I))
024          AME=A(I)-E
024          SUM2=SUM2+AME*AME
024     141  CONTINUE
024          CALL CONLIM (E,FM,ROOTM,SUM2)
024          GO TO 149
024     142  PRINT 32
024          GO TO 149
024    C         COMPUTE FOR EACH SITE.
024     144  DO 148 IS=1,NS
024          NN=KK(IS)
024          IF (NN.GT.0) GO TO 145
024          ERR(IS)=-1.0
024          GO TO  148
024     145  CALL TEST1 (NN,SOCC(1,IS),SPCC(1,IS),E)
024          ERR(IS)=E
024          WRITE (OUT,11) NOS(IS),E
024          IF (LM(IS).LT.NCRIT) GO TO 147
024    C       DETERMINE THE ERROR AND C.L.
024          SUM2=0.0
024          DO 146 I=1,M
024          CALL TEST1 (NODS(I,IS),SOCA(1,I,IS),SPCA(1,I,IS),A(I))
024          AME=A(I)-E
024          SUM2=SUM2+AME*AME
024     146  CONTINUE
024          CALL CONLIM (E,FM,ROOTM,SUM2)
025          IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024          GO TO 148
024     147  PRINT 32
025          IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024     148  CONTINUE
025          IF (MOD(NS,5).NE.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024          IF (IMAP.EQ.3HNO ) GO TO 149
024          ENORM(1)=-1.0
024          CALL RAMMAP (1,ERR,ENORM)
024     149  IF (MODE.EQ.2) GO TO 150
024          GO TO 290
024    C
024    C          2.  COMPUTE THE MEAN SQUARED ERROR.
024    C
024     150  IF (MODE.EQ.2.AND.NOT(2).EQ.0) GO TO 161
024          IF (ICODE.LT.0) GO TO 154
024    C          COMPUTE FOR ONE/ALL SITES.
024          CALL TEST2 (N,OC,PC,E)
```

```
024          F=SQRT(E)
024          WRITE (OUT,13) NOS(ISN),E
024          IF (LN.LT.NCRIT) GO TO 152
024    C          DETERMINE THE ERROR AND C.L.
024          SUM2=0.0
024          DO 151 I=1,M
024          CALL TEST2 (NODS(I,IS),OCA(1,I),PCA(1,I),A(I))
024          AME=A(I)-E
024          SUM2=SUM2+AME*AME
024     151  CONTINUE
024          CALL CONLIM (E,FM,ROOTM,SUM2)
024          PRINT 37,F
024          GO TO 159
024     152  PRINT 32
024          PRINT 37,F
024          GO TO 159
024    C          COMPUTE FOR EACH SITE.
024     154  DO 158 IS=1,NS
024          NN=KK(IS)
024          IF (NN.GT.0) GO TO 155
024          ERR(IS)=-1.0
024          GO TO 158
024     155  CALL TEST2 (NN,SOCC(1,IS),SPCC(1,IS),E)
024          ERR(IS)=E
024          F=SQRT(E)
024          WRITE (OUT,13) NOS(IS),E
024          IF (LM(IS).LT.NCRIT) GO TO 157
024    C         DETERMINE THE ERROR AND C.L.
024          SUM2=0.0
024          DO 156 I=1,M
024          CALL TEST2 (NODS(I,IS),SOCA(1,I,IS),SPCA(1,I,IS),A(I))
024          AME=A(I)-E
024          SUM2=SUM2+AME*AME
024     156  CONTINUE
024          CALL CONLIM (E,FM,ROOTM,SUM2)
024          PRINT 37,F
025          IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024          GO TO 158
024     157  PRINT 32
024          PRINT 37,F
025          IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024     158  CONTINUE
025          IF (MOD(NS,5).NE.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024          IF (IMAP.EQ.3HNO ) GO TO 159
024          ENORM(1)=-1.0
024          CALL RAMMAP (2,ERR,ENORM)
024     159  IF (MODE.EQ.2) GO TO 161
024          GO TO 290
024    C
024    C           3.  COMPUTE THE ABSOLUTE ERROR THRESHOLD.
024    C
024     160  WRITE (OUT,15)
024     161  IF (MODE.EQ.2.AND.NOT(3).EQ.0) GO TO 171
025          WRITE (OUT,41)
024          READ (INP,3) E1
024          ENORM(1)=E1
024          IF (ICODE.LT.0) GO TO 164
```

```
024   C           COMPUTE FOR ONE/ALL SITES.
024         CALL TEST3 (N,OC,PC,E)
024         WRITE (OUT,16) NOS(ISN),E
024         IF (LN.LT.NCRIT) GO TO 163
024   C           DETERMINE THE ERROR AND C.L.
024         SUM2=0.0
024         DO 162 I=1,M
024         CALL TEST3 (NODS(I,IS),OCA(1,I),PCA(1,I),A(I))
024         AME=A(I)-E
024         SUM2=SUM2+AME*AME
024     162 CONTINUE
024         CALL CONLIM (E,FM,ROOTM,SUM2)
024         GO TO 169
024     163 PRINT 32
024         GO TO 169
024   C           COMPUTE FOR EACH SITE.
024     164 DO 168 IS=1,NS
024         NN=KK(IS)
024         IF (NN.GT.0) GO TO 165
024         ERR(IS)=-1.0
024         GO TO 168
024     165 CALL TEST5 (NN,SOCC(1,IS),SPCC(1,IS),E)
024         ERR(IS)=E
024         WRITE (OUT,16) NOS(IS),E
024         IF (LM(IS).LT.NCRIT) GO TO 167
024   C           DETERMINE THE ERROR AND C.L.
024         SUM2=0.0
024         DO 166 I=1,M
024         CALL TEST3 (NODS(I,IS),SOCA(1,I,IS),SPCA(1,I,IS),A(I))
024         AME=A(I)-E
024         SUM2=SUM2+AME*AME
024     166 CONTINUE
024         CALL CONLIM (E,FM,ROOTM,SUM2)
025         IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
025         GO TO 168
024     167 PRINT 32
025         IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024     168 CONTINUE
025         IF (MOD(NS,5).NE.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024         IF (IMAP.EQ.3HNO ) GO TO 169
024         CALL RAMMAP (3,ERR,ENORM)
024     169 IF (MODE.EQ.2) GO TO 171
024         WRITE (OUT,17)
024         READ (INP,18) IYON
024         IF (IYON.EQ.3HYES) GO TO 160
024         GO TO 290
024   C
024   C            4.  COMPUTE THE PERCENTAGE ERROR THRESHOLD.
024   C
024     170 WRITE (OUT,19)
024     171 IF (MODE.EQ.2.AND.NOT(4).EQ.0) GO TO 161
025         WRITE (OUT,42)
024         READ 3,E2
024         ENORM(1)=E2
024         IF (ICODE.LT.0) GO TO 174
024   C           COMPUTE FOR ONE/ALL SITES.
024         CALL TEST4 (N,OC,PC,E)
```

```
024              WRITE (OUT,20) NOS(ISN),E
024              IF (LN.LT.NCRIT) GO TO 173
024    C               DETERMINE THE ERROR AND C.L.
024              SUM2=0.0
024              DO 172 I=1,M
024              CALL TEST4 (NODS(I,IS),OCA(1,I),PCA(1,I),A(I))
024              AME=A(I)-E
024              SUM2=SUM2+AME*AME
024      172     CONTINUE
024              CALL CONLIM (E,FM,ROOTM,SUM2)
024              GO TO 179
024      173     PRINT 32
024              GO TO 179
024    C              COMPUTE FOR EACH SITE.
024      174     DO 178 IS=1,NS
024              NN=KK(IS)
024              IF (NN.GT.0) GO TO 175
024              ERR(IS)=-1.0
024              GO TO 178
024      175     CALL TEST4 (NN,SOCC(1,IS),SPCC(1,IS),E)
024              ERR(IS)=E
024              WRITE (OUT,20) NOS(IS),E
024              IF (LM(IS).LT.NCRIT) GO TO 177
024    C               DETERMINE THE ERRCR AND C.L.
024              SUM2=0.0
024              DO 176 I=1,M
024              CALL TEST4 (NODS(I,IS),SOCA(1,I,IS),SPCA(1,I,IS),A(I))
024              AME=A(I)-E
024              SUM2=SUM2+APE*AME
024      176     CONTINUE
024              CALL CONLIM (E,FM,ROOTM,SUM2)
025              IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024              GO TO 178
024      177     PRINT 32
025              IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024      178     CONTINUE
025              IF (MOD(NS,5).NE.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024              IF (IMAP.EQ.3HNO ) GO TO 179
024              CALL RAMMAP (4,ERR,ENORM)
024      179     IF (MODE.EQ.2) GO TO 181
024              WRITE (OUT,17)
024              READ (INP,18) IYON
024              IF (IYON.EQ.3HYES) GO TO 170
024              GO TO 290
024    C
024    C               5.  COMPUTE THE SYMMETRIC HIGH-LOW LOSS FUNCTION.
024    C
024      180     WRITE (OUT,21)
024      181     IF (MODE.EQ.2.AND.NOT(5).EQ.0) GO TO 191
025              WRITE (OUT,43)
024              READ (INP,3) E3
024              ENORM(1)=E3
024              IF (ICODE.LT.0) GO TO 184
024    C               COMPUTE FOR ONE/ALL SITES.
024              CALL TEST5 (N,OC,PC,E)
024              WRITE (OUT,22) NOS(ISN),E
024              IF (LN.LT.NCRIT) GO TO 183
```

```
C00512     024   C          DETERMINE THE ERROR AND C.I.
C00513     024           SUM2=0.0
C00514     024           DO 182 I=1,M
C00515     024           CALL TESTS (NODS(I,IS),OCA(1,I),PCA(1,I),A(I))
C00516     024           AMI=A(I)-E
C00517     024           SUM2=SUM2+AMI*AMI
C00518     024     182   CONTINUE
C00519     024           CALL CONLIM (E,EM,ROOTM,SUM2)
C00520     024           GO TO 189
C00521     024     183   PRINT 32
C00522     024           GO TO 189
C00523     024   C          COMPUTE FOR EACH SITE.
C00524     024     184   DO 188 IS=1,NS
C00525     024           NN=KK(IS)
C00526     024           IF (NN.GT.0) GO TO 185
C00527     024           ERR(IS)=-1.0
C00528     024           GO TO 188
C00529     024     185   CALL TESTS (NN,SOCC(1,IS),SPCC(1,IS),E)
C00530     024           ERR(IS)=E
C00531     024           WRITE (OUT,22) NOS(IS),E
C00532     024           IF (EM(IS).LT.NCRIT) GO TO 187
C00533     024   C          DETERMINE THE ERROR AND C.I.
C00534     024           SUM2=0.0
C00535     024           DO 186 I=1,M
C00536     024           CALL TESTS (NODS(I,IS),SOCA(1,I,IS),SPCA(1,I,IS),A(I))
C00537     024           AMI=A(I)-E
C00538     024           SUM2=SUM2+AMI*AMI
C00539     024     186   CONTINUE
C00540     024           CALL CONLIM (E,EM,ROOTM,SUM2)
C00541     025           IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
C00542     024           GO TO 188
C00543     024     187   PRINT 32
C00544     025           IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
C00545     024     188   CONTINUE
C00546     025           IF (MOD(NS,5).NE.0.AND.IMAP.EQ.3HYES) CALL HCOPY
C00547     024           IF (IMAP.EQ.3HNO ) GO TO 189
C00548     024           CALL RAMMAP (5,ERR,ENORM)
C00549     024     189   IF (MODE.EQ.2) GO TO 191
C00550     024           WRITE (OUT,17)
C00551     024           READ (INP,18) IYON
C00552     024           IF (IYON.EQ.3HYES) GO TO 180
C00553     024           GO TO 290
C00554     024   C
C00555     024   C          6.  COMPUTE THE ASYMMETRIC HIGH-LOW LOSS FUNCTION.
C00556     024   C
C00557     024     190   WRITE (OUT,23)
C00558     024     191   IF (MODE.EQ.2.AND.NOT(2).EQ.0) GO TO 201
C00559     025           WRITE (OUT,44)
C00560     024           READ (INP,3) E4,HE1,HE2
C00561     024           ENORM(1)=E4
C00562     024           ENORM(2)=HE1
C00563     024           ENORM(3)=HE2
C00564     024           IF (ICODE.LT.9) GO TO 194
C00565     024   C          COMPUTE FOR ONE/ALL SITES.
C00566     024           CALL TEST6 (N,OC,PC,E)
C00567     024           WRITE (OUT,24) NOS(ISN),E
C00568     024           IF (EN.LT.NCRIT) GO TO 193
```

Reproduced from best available copy.

```
024   C          DETERMINE THE ERROR AND C.L.
024              SUM2=0.0
024              DO 192 I=1,M
024              CALL TEST6 (NODS(I,IS),OCA(1,I),PCA(1,I),A(I))
024              AME=A(I)-E
024              SUM2=SUM2+AME*AME
024        192   CONTINUE
024              CALL CONLIM (E,FM,ROOTM,SUM2)
024              GO TO 199
024        193   PRINT 32
024              GO TO 199
024   C             COMPUTE FOR EACH SITE.
024        194   DO 198 IS=1,NS
024              NN=KK(IS)
024              IF (NN.GT.0) GO TO 195
024              ERR(IS)=-1.0
024              GO TO 198
024        195   CALL TEST6 (NN,SOCC(1,IS),SPCC(1,IS),E)
024              ERR(IS)=E
024              WRITE (OUT,24) NOS(IS),E
024              IF (LM(IS).LT.NCRIT) GO TO 197
024   C             DETERMINE THE ERROR AND C.L.
024              SUM2=0.0
024              DO 196 I=1,M
024              CALL TEST6 (NODS(I,IS),SOCA(1,I,IS),SPCA(1,I,IS),A(I))
024              AME=A(I)-E
024              SUM2=SUM2+AME*AME
024        196   CONTINUE
024              CALL CONLIM (E,FM,ROOTM,SUM2)
025              IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024              GO TO 198
024        197   PRINT 32
025              IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024        198   CONTINUE
025              IF (MOD(NS,5).NE.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024              IF (IMAP.EQ.3HNO ) GO TO 199
024              CALL RAMMAP (6,ERR,ENORM)
024        199   IF (MODE.EQ.2) GO TO 201
024              WRITE (OUT,17)
024              READ (INP,18) IYON
024              IF (IYON.EQ.3HYES) GO TO 190
024              GO TO 290
024   C
024   C             7.  DETERMINE THE ERROR FROM A USER SUPPLIED LOSS MATRIX BASE
024   C          UPON PC AND OC.
024   C
024        200   WRITE (OUT,30)
024              READ (INP,18) IYON
024              IF (IYON.EQ.3HYES) GO TO 210
024   C             QUICK-RUN USES PROGRAM LOSS-MATRIX.
025        201   IF (MODE.EQ.2.AND.NOT(7).EQ.0) GO TO 225
024              DO 205 I=1,5
024              DO 205 J=1,5
024              AS(I,J)=BS(I,J)
024        205   CONTINUE
025              IF (MODE.EQ.1) GO TO 215
025              WRITE (OUT,45)
```

```
025             GO TO 216
024       210   READ (INP,3) ((AS(I,J),J=1,5),I=1,5)
024             WRITE (OUT,33) ((AS(I,J),J=1,5),I=1,5)
024             WRITE (OUT,34)
024             READ (INP,18) IYON
024             IF (IYON.EQ.3HNO ) GO TO 200
025       215   WRITE (OUT,25)
024       216   READ (INP,3) CMAX
024             ENORM(1)=CMAX
024             IF (ICODE.LT.0) GO TO 219
024    C             COMPUTE FOR ONE/ALL SITES.
024             CALL TEST7 (N,OC,PC,E)
024             WRITE (OUT,26) NOS(ISN),E
024             IF (LN.LT.NCRIT) GO TO 218
024    C             DETERMINE THE ERROR AND C.L.
024             SUM2=0.0
024             DO 217 I=1,M
024             CALL TEST7 (NODS(I,IS),OCA(1,I),PCA(1,I),A(I))
024             AME=A(I)-E
024             SUM2=SUM2+AME*AME
024       217   CONTINUE
024             CALL CONLIM (E,FM,ROOTM,SUM2)
024             GO TO 224
024       218   PRINT 32
024             GO TO 224
024    C              COMPUTE FOR EACH SITE.
024       219   DO 223 IS=1,NS
024             NN=KK(IS)
024             IF (NN.GT.0) GO TO 220
024             ERR(IS)=-1.0
024             GO TO 223
024       220   CALL TEST7 (NN,SOCC(1,IS),SPCC(1,IS),E)
024             ERR(IS)=E
024             WRITE (OUT,26) NOS(IS),E
024             IF (LM(IS).LT.NCRIT) GO TO 222
024    C              DETERMINE THE ERROR AND C.L.
024             SUM2=0.0
024             DO 221 I=1,M
024             CALL TEST7 (NODS(I,IS),SOCA(1,I,IS),SPCA(1,I,IS),A(I))
024             AME=A(I)-E
024             SUM2=SUM2+AME*AME
024       221   CONTINUE
024             CALL CONLIM (E,FM,ROOTM,SUM2)
025             IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024             GO TO 223
024       222   PRINT 32
025             IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024       223   CONTINUE
025             IF (MOD(NS,5).NE.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024             IF (IMAP.EQ.3HNO ) GO TO 224
024             CALL RAMMAP (7,ERR,ENORM)
025       224   IF (MODE.EQ.2) GO TO 225
024             WRITE (OUT,29)
024             READ (INP,18) IYON
024             IF (IYON.EQ.3HYES) GO TO 215
024             WRITE (OUT,31)
024             READ (INP,18) IYON
```

```
024          IF (IYON.EQ.3HYES) GO TO 200
024   C
024   C           8.  DETERMINE THE LOCATION WHERE THE MAX CONCENTRATION OCCURS
024   C
025    225   IF (ICODE.EQ.-1.AND.NOT(8).EQ.1) GO TO 226
025          IF (NOT(8).EQ.0) GO TO 280
025          WRITE (OUT,46)
025          GO TO 280
025    226   IF (MODE.EQ.2) GO TO 227
025          WRITE (OUT,27)
024          READ (INP,18) IYON
025          IF (IYON.EQ.3HYES) GO TO 229
025   C           QUICK-RUN USES PROGRAM DISTANCE MATRIX.
025    227   DO 228 I=1,NS
025          DO 228 J=1,NS
024          DD(I,J)=DS(I,J)
025    228   CONTINUE
025          IF (MODE.EQ.1) GO TO 230
025          WRITE (OUT,47)
024          GO TO 230
025    229   READ (INP,3) ((DD(J,K),K=1,NS),J=1,NS)
024           WRITE (OUT,35) ((DD(J,K),K=1,NS),J=1,NS)
024          WRITE (OUT,34)
024          READ (INP,18) IYON
024           IF (IYON.EQ.3HNO ) GO TO 225
024    230   IF (ICODE.LT.0) GO TO 250
024   C          COMPUTE FOR ONE/ALL SITES.
024          CALL TEST8 (N,OC,PC,E)
024          WRITE (OUT,28) NOS(ISN),E
024          IF (LN.LT.NCRIT) GO TO 240
024   C          DETERMINE THE ERROR AND C.L.
024          SUM2=0.0
024          DO 235 I=1,M
024          CALL TEST8 (NODS(I,IS),OCA(1,I),PCA(1,I),A(I))
024          AME=A(I)-E
024          SUM2=SUM2+AME*AME
024    235   CONTINUE
024          CALL CONLIM (E,FM,ROOTM,SUM2)
024          GO TO 280
024    240   PRINT 32
024          GO TO 280
024   C           COMPUTE FOR EACH SITE.
024    250   DO 265 IS=1,NS
024          NN=KK(IS)
024          IF (NN.EQ.0) GO TO 265
024          CALL TEST8 (NN,SOCC(1,IS),SPCC(1,IS),E)
024          WRITE (OUT,28) NOS(IS),E
024          IF (LM(IS).LT.NCRIT) GO TO 260
024   C           DETERMINE THE ERROR AND C.L.
024          SUM2=0.0
024          DO 255 I=1,M
024          CALL TEST8 (NODS(I,IS),SOCA(1,I,IS),SPCA(1,I,IS),A(I))
024          AME=A(I)-E
024          SUM2=SUM2+AME*AME
024    255   CONTINUE
024          CALL CONLIM (E,FM,ROOTM,SUM2)
025          IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
```

```
024          GO TO 265
024     260  PRINT 32
025          IF (MOD(IS,5).EQ.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024     265  CONTINUE
025          IF (MOD(NS,5).NE.0.AND.IMAP.EQ.3HYES) CALL HCOPY
024     280  IF (MODE.EQ.2) GO TO 300
024   C           TRY ANOTHER TEST?
024     290  WRITE (OUT,12)
024          READ (INP,18) IYON
024          IF (IYON.EQ.3HYES) GO TO 130
024   C           TRY ANOTHER MODE?
024     300  WRITE (OUT,36)
024          READ (INP,18) IYON
025          IF (IYON.EQ.3HYES) GO TO 99
024   C           TRY ANOTHER SORT?
024          WRITE (OUT,38)
024          READ (INP,18) IYON
025          IF (IYON.EQ.3HYES) GO TO 54
024          WRITE (OUT,86)
025          IF (IMAP.EQ.3HYES) CALL FINITT (0,20)
024          STOP
024          END

END ELT.
```

C.2 Program Map, M21ADO*STAT01.MAPIT

```
@ELT,L S1.MAPIT
ELT007 SL73R1 11/01/78 09:38:21 (4,)
C00001        003    @MAP ,M21ADO*STAT01.ACSCOR
C00002        001    IN M21ADO*STAT01.ACSCOR
C00003        001    IN M21ADO*STAT01.CONLIM
C00004        001    IN M21ADO*STAT01.TESTS
C00005        001    IN SRI*SRI.RAMMAP
C00006        001    IN SRI*SRI.BOX
C00007        001    IN SRI*SRI.HEADER
C00008        001    IN RAPS*UTILITY.COMPOZ
C00009        001    LIB GRAPH*TEKTRONIX.
C00010        001    END
C00011        003    @XQT M21ADO*STAT01.ACSCOR

END ELT.
```

C.3 Subroutine M21ADO*STAT01.TESTS

```
@ELT,L S1.TESTS
ELT007 SL73R1 11/01/78 09:38:22 (1,)
000001        000          SUBROUTINE TESTS (N,OC,PC,E)
000002        000          DIMENSION OC(N),PC(N)
000003        000          COMMON /DAT/E1,E2,E3,E4,PCMIN,HL1,HL2,CMAX,DD(5,5),AS(5,5)
000004        000          COMMON /VAR/IDATE(123),ITIME(123),ISITE(123)
000005        000          RETURN
000006        000          ENTRY TEST1 (N,OC,PC,E)
000007        000          SUM=0.0
000008        000          DO 100 I=1,N
000009        000          SUM=SUM+ABS(OC(I)-PC(I))
000010        000      100 CONTINUE
000011        000          E=SUM/N
000012        000          RETURN
000013        000          ENTRY TEST2 (N,OC,PC,E)
000014        000          SUM=0.0
000015        000          DO 110 I=1,N
000016        000          OCMPC=OC(I)-PC(I)
000017        000          SUM=SUM+OCMPC*OCMPC
000018        000      110 CONTINUE
000019        000          E=SUM/N
000020        000          RETURN
000021        000          ENTRY TEST3 (N,OC,PC,E)
000022        000          SUM=0.0
000023        000          DO 120 I=1,N
000024        000          IF (ABS(OC(I)-PC(I)).GT.E1) SUM=SUM+1.0
000025        000      120 CONTINUE
000026        000          E=SUM/N
000027        000          RETURN
000028        000          ENTRY TEST4 (N,OC,PC,E)
000029        000          SUM=0.0
000030        000          DO 130 I=1,N
000031        000          IF (OC(I).GT.0.0) GO TO 125
000032        000          IF (PC(I).LE.PCMIN) GO TO 130
000033        000          SUM=SUM+1.0
000034        000          GO TO 130
000035        000      125 IF (ABS((OC(I)-PC(I))/OC(I)).GT.E2) SUM=SUM+1.0
000036        000      130 CONTINUE
000037        000          E=SUM/N
000038        000          RETURN
000039        000          ENTRY TEST5 (N,OC,PC,E)
000040        000          SUM=0.0
000041        000          DO 140 I=1,N
000042        000          IF (OC(I).LE.E3.AND.PC(I).LE.E3) GO TO 140
000043        000          IF (OC(I).GT.E3.AND.PC(I).GT.E3) GO TO 140
000044        000          SUM=SUM+1.0
000045        000      140 CONTINUE
000046        000          E=SUM/N
000047        000          RETURN
000048        000          ENTRY TEST6 (N,OC,PC,E)
000049        000          SUM=0.0
000050        000          DO 150 I=1,N
000051        000          IF (OC(I).LT.E4.AND.PC(I).GT.E4) SUM=SUM+HL1
000052        000          IF (OC(I).GT.E4.AND.PC(I).LT.E4) SUM=SUM+HL2
000053        000      150 CONTINUE
000054        000          E=SUM/N
000055        000          RETURN
```

```
000            ENTRY TEST7 (N,OC,PC,E)
000            SUM=0.0
000            DO 160 I=1,N
000            J=5.0*((PC(I)+0.2*CMAX)/CMAX)
000            IF (J.GT.5) K=5
000            K=5.0*((OC(I)+0.2*CMAX)/CMAX)
000            IF (K.GT.5) K=5
000            SUM=SUM+AS(J,K)
000      160   CONTINUE
000            E=SUM/N
000            RETURN
000            ENTRY TEST8 (N,OC,PC,E)
000            SUM=0.0
000            NT=0
000            MOC=1
000            MPC=1
000            OMAX=-1.0E10
000            PMAX=-1.0E10
000            IDATED=IDATE(1)
000            ITIMED=ITIME(1)
000            DO 185 I=1,N
000            IF (IDATE(I).NE.IDATED) GO TO 180
000            IF (ITIME(I).NE.ITIMED) GO TO 180
000      170   IF (OC(I).LT.OMAX) GO TO 175
000            OMAX=OC(I)
000            MOC=ISITE(I)
000      175   IF (PC(I).LT.PMAX) GO TO 185
000            PMAX=PC(I)
000            MPC=ISITE(I)
000            GO TO 185
000      180   NT=NT+1
000            SUM=SUM+DD(MOC,MPC)
000            MOC=1
000            MPC=1
000            OMAX=-1.0E10
000            PMAX=-1.0E10
000            IDATED=IDATE(I)
000            ITIMED=ITIME(I)
000            GO TO 170
000      185   CONTINUE
000            E=SUM/NT
000            RETURN
000            END

END ELT.

@HDG,N
```

C.3 Subroutine M21ADO*STAT01.CONLIM

```
@ELT,L S1.CONLIM
ELT007 SL73R1 11/01/78 09:38:08 (2,)
000001       000          SUBROUTINE CONLIM (E,FM,ROOTM,SUM2)
000002       000          S=SQRT(SUM2/(FM-1.0))
000003       000          CLU=1.96*S/ROOTM
000004       000          CLL=-CLU
000005       000          CL1=E+CLU
000006       000          CL2=E+CLL
000007       000          IF (CL2.LT.0) CL2=0.0
000008       000          PRINT 40,CL2,CL1
000009       000      40   FORMAT (/12X24H95% CONFIDENCE INTERVAL:E10.3,4H TO E10.3)
000010       000          RETURN
000011       000          END

END ELT.
```

C.3 Subroutine M21ADO*STAT01.RAMMAP

```
@ELT,L S.RAMMAP
ELT007 SL73R1 11/01/78 09:37:10 (30,)
       026         SUBROUTINE RAMMAP(NTEST,VALUES,PARM)
       026   C ROD ALLEN, COMP-AID INC., OCTOBER 1978
       026   C PLOT VALUES ON RAMS STATION MAP OF ST. LOUIS FOR RAM MODEL TESTS
       026   C FOR RON RUFF OF SRI
       026   C ASSUMES TEKTRONIX 4014
       026   C 3 E1 THRESHOLD, 4 E2 THRESHOLD, 5 E3 NORM, 6 E4 NORM, L1 LOSS 1, L2 LOSS 2,
       026   C 7 LOSS MATRIX SUPPLIED BY USER, NOT LISTED HERE
       026   C DECLARATIONS
       026         DIMENSION TEST(6,7),ZERO(4),VALUES(25),LOC(25),PARM(3),AOUT(5)
       026         DIMENSION RAMSX(25),RAMSY(25),XLAB(4),YLAB(4),NUMBER(25)
       026         DIMENSION MOX(5),MOY(5),MSX(14),MSY(14),MSLX(6),MSLY(6)
       026   C INITIALIZATION OF DATA
       026         DATA MMS/14/,MMO/5/,MSL/6/
       026         DATA MOX/696,710,720,732,750/,MOY/4276,4284,4297,4306,4300/
       026         DATA MSLX/739,735,733,734,738,746/,MSLY/4269,4270,4275,4282,4288,
       026        1 4294/
       026         DATA MSX/730,732,739,744,746,744,744,746,750,750,735,723,714,708/
       026         DATA MSY/4235,4251,4268,4273,4280,4286,4290,4294,4300,4306,4310,
       026        1 4314,4314,4330/
       026         DATA RAMSX/744.2,742.5,747.6,747.3,743.7,
       026        1 738.7,740.2,748.4,755.8,747.2, 738.8,733.9,737.7,744.3,757.1,
       026        2 762.8,760.6,743.1,729.8,723.1, 732.4,741.6,777.3,749.3,697.4/
       026         DATA RAMSY/4280.,4286.,4282.,4277.,4276.,
       026        1 4278.,4283.,4291.,4280.,4273., 4272.,4281.,4290.,4297.,4298.,
       026        2 4290.,4273.,4263.,4271.,4286., 4302.,4329.,4286.,4237.,4282./
       026         DATA XLAB/'700','720','740','760'/
       026         DATA NUMBER/'01','02','03','04','05','06','07','08','09','10',
       026        1 '11','12','13','14','15','16','17','18','19','20','21','22',
       026        2 '23','24','25'/
       026         DATA LOC/1,1,1,1,3,3,3,1,1,1,4,3,3,3,1,1,1,1,3,3,1,4,3,2,1/
       026         DATA ZERO/'0.',' 0.','  0.',' 0.'/
       026         DATA YLAB/'4250','4275','4300','4325'/
       026         DATA TEST/'MEAN ABSOLUTE ERROR                 ',
       026        2          'MEAN SQUARE ERROR                   ',
       026        3          'ABSOLUTE ERROR THRESHOLD            ',
       026        4          'PERCENTAGE ERROR THRESHOLD          ',
       026        5          'SYMMETRIC HIGH-LOW LOSS FUNCTION    ',
       026        6          'ASYMMETRIC HIGH-LOW LOSS FUNCTION   ',
       026        7          'USER SUPPLIED LOSS MATRIX           '/
       026   C INITIALIZATION
       028         WRITE(6,75)
       028   75    FORMAT(/,' PRESS RETURN TO CONTINUE')
       028         READ 76, I
       028   76    FORMAT(A6)
       026         CALL INITT(120)
       026         CALL TERM(2,1024)
       026   C WRITE TITLE WHICH IS TEST DESCRIPTION OF TEST NTEST
       026         CALL CHRSIZ(2)
       026         CALL MOVABS(20,700)
       026         CALL AOUTST(5,'TEST ')
       026         N=NUMBER(NTEST)*64
       026         CALL AOUTST(1,N)
       026         CALL MOVABS(20,680)
       026         CALL AOUTST(33,TEST(1,NTEST))
       026   C PRINT PARAMETERS
```

```
026          IF(NTEST.LE.2) GO TO 80
026          CALL MOVABS(20,660)
026          IF(NTEST.LE.6) GO TO 81
026          CALL AOUTST(45,"LOSS MATRIX SUPPLIED BY USER, NOT LISTED HERE")
029          CALL MOVABS(20,640)
026    81    ENCODE(30,82,AOUT) PARM
026    82    FORMAT(1PG8.2,4X,1P2G8.2,2X)
026          IF(NTEST.EQ.3) CALL AOUTST(12,"E1 THRESHOLD")
026          IF(NTEST.EQ.4) CALL AOUTST(12,"E2 THRESHOLD")
026          IF(NTEST.EQ.5) CALL AOUTST(7,"E3 NORM")
026          IF(NTEST.EQ.6) CALL AOUTST(25,"E4 NORM, L1 LOSS, L2 LOSS")
029          IF(NTEST.EQ.7) CALL AOUTST(26,"CMAX CONCENTRATION MAXIMUM")
026          CALL AOUTST(8,AOUT)
026          IF(NTEST.EQ.6) CALL AOUTST(16,AOUT(3))
026    80    CALL CHRSIZ(3)
026    C USES THE NEXT TO SMALLEST OF 4 CHARACTER SIZES
026    C WRITE SUBTITLES AND LEGEND
026          CALL MOVABS(20,200)
026          CALL AOUTST(29,"VALUES GIVEN NEXT TO STATIONS")
026          CALL MOVABS(22,184)
026          CALL AOUTST(15,"SN RAMS STATION")
026          CALL BOX(20,182,38,196)
026          CALL MOVABS(20,168)
026          CALL AOUTST(34,"WHERE SN+100 IS THE STATION NUMBER")
026          CALL MOVABS(20,152)
026          CALL AOUTST(37,"VALUE EXAMPLES 12-3=.012 AND 12+3=120")
026          CALL MOVABS(20,136)
026          CALL AOUTST(26,"AXES UNITS UTM COORDINATES")
026    C DRAW FRAME
026          CALL BOX(0,0,1023,780)
026    C DRAW TIC MARKS
026          DO 1 I=1,4
026          IX=1024*(680.+20*I-696.3)/(778.4-696.3)
026          IY=781*(4225.+25*I-4235.7)/(4330.3-4235.7)
026          CALL MOVABS(0,IY)
026          CALL DRWABS(4,IY)
026          CALL MOVABS(1023,IY)
026          CALL DRWABS(1019,IY)
026          CALL MOVABS(IX,0)
026          CALL DRWABS(IX,4)
026          CALL MOVABS(IX,776)
026          CALL DRWABS(IX,780)
026          CALL MOVABS(IX-8,5)
026          CALL AOUTST(3,XLAB(I))
026          CALL MOVABS(5,IY-6)
026    1     CALL AOUTST(4,YLAB(I))
026    C PRINT TIME AND DATE HEADER
026          CALL HEADER
026    C DRAW STATION BOXES AND WRITE STATION NUMBERS
026          DO 2 I=1,25
026          IX=1024*(RAMSX(I)-696.3)/(778.4-696.3)
026          IY=781*(RAMSY(I)-4235.7)/(4330.3-4235.7)
026          CALL BOX(IX-9,IY-7,IX+9,IY+7)
026          CALL MOVABS(IX-7,IY-5)
026          CALL AOUTST(2,NUMBER(I))
026    C WRITE VALUES NEAR STATIONS
026          IF(LOC(I).EQ.1) CALL MOVABS(IX+13,IY-5)
```

```
026          IF(LOC(I).EQ.2) CALL MUVAPS(IX-15,IY+11)
026          IF(LOC(I).EG.3) CALL MGVAPS(IX-43,IY-5)
026          IF(LOC(I).EQ.4) CALL MOVAFS(IX-15,IY-20)
026    C ENCODE OUTPUT OF VALUES(I)
027    C IF VALUES(I) < 0 LEAVE BLANK
027          IF(VALUES(I)) 2,6,3
027    6     L=LOC(I)
026          CALL AOUTST(4,ZERO(L))
026          GO TO 2
026    C IF VALUES(I) > 99+09 OR <= 99-10 USE X+XX FORMAT
026    C IF VALUES(I) <= 99+09 AND > 99-10 USE XX+X FORMAT
026    3     IF(VALUES(I).GT.99.E+9.OR.VALUES(I).LE.99.E-10) GO TO 4
026          ENCODE(6,100,IVAL) VALUES(I)
026    100   FORMAT(2PE6)
026          CALL AOUTST(2,IVAL)
026    C SHIFT LEFT DECIMAL POINT AND PRINT SIGN
026          IVAL=IVAL*64*64*64
026          CALL AOUTST(1,IVAL)
026    C SHIFT LEFT 0 FROM EXPONENT AND PRINT UNIT PLACE OF EXPONENT
026          IVAL=IVAL*64*64
026          CALL AOUTST(1,IVAL)
026          GO TO 2
026    4     ENCODE(5,101,IVAL) VALUES(I)
026    101   FORMAT(1PE5)
026          CALL AOUTST(1,IVAL)
026    C SHIFT LEFT DECIMAL POINT AND PRINT SIGN AND EXPONENT
026          IVAL=IVAL*64*64
026          CALL AOUTST(3,IVAL)
026    2     CONTINUE
026    C PLOT RIVERS AND CITY
026          CALL MOVABS(100,480)
026          CALL CHRSIZ(3)
026          CALL AOUTST(14,"MISSOURI RIVER")
026          CALL MOVABS(460,100)
026          CALL AOUTST(17,"MISSISSIPPI RIVER")
026          CALL MOVABS(615,430)
026          CALL AOUTST(17,"CITY OF ST. LOUIS")
026          DO 40 I=1,MPS
026          IX=1024*(MSX(I)-696.3)/(778.4-696.3)
026          IY=781*(MSY(I)-4235.7)/(4330.3-4235.7)
026          IF(I.EQ.1) CALL MOVABS(IX,IY)
026    40    CALL DRWABS(IX,IY)
026          DO 41 I=1,MMO
026          IX=1024*(MOX(I)-696.3)/(778.4-696.3)
026          IY=781*(MOY(I)-4235.7)/(4330.3-4235.7)
026          IF(I.EQ.1) CALL MOVABS(IX,IY)
026    41    CALL DRWABS(IX,IY)
026          DO 42 I=1,MSL
026          IX=1024*(MSLX(I)-696.3)/(778.4-696.3)
026          IY=781*(MSLY(I)-4235.7)/(4330.3-4235.7)
026          IF(I.EQ.1) CALL MOVABS(IX,IY)
026    42    CALL DRWABS(IX,IY)
026    C TERMINATION
030    C
030          ENTRY HCOPY
030    C
026          CALL HDCOPY
```

```
000170        027          CALL MOVABS(10,50)
000171        027          CALL AOUTST(24,"PRESS RETURN TO CONTINUE")
000172        027          CALL BELL
000173        027          CALL TSEND
000174        027          CALL TINPUT(I)
000175        027          CALL ERASE
000176        030    C     CALL FINITT(0,780)
000177        026          RETURN
000178        026          END

END ELT.
```

C.3 Subroutine M21ADO*STAT01.HEADER

```
@ELT,L S.HEADER
ELT007 SL73R1 11/01/78 09:36:20 (18,)
000001       016          SUBROUTINE HEADER
000002       014    C ROD ALLEN, COMP-AID INC., OCTOBER 1978
000003       016    C PRINT TIME AND DATE HEADER IN UPPER RIGHT CORNER AND BOX IT
000004       016    C ASSUMES TEKTRONIX 4014
000005       017          DIMENSION DATE(2),TIME(2)
000006       018          CALL CHRSIZ(3)
000007       014          CALL BOX(953,738,1023,780)
000008       014          CALL COMPOZ(0,IL,DATE)
000009       014          CALL COMPOZ(18)
000010       014          CALL COMPOZ(0,IL,TIME)
000011       014          CALL COMPOZ(17)
000012       014          CALL MOVABS(956,767)
000013       014          CALL AOUTST(7,"PLOTTED")
000014       014          CALL MOVABS(956,754)
000015       014          CALL AOUTST(8,DATE)
000016       014          CALL MOVABS(956,741)
000017       014          CALL AOUTST(8,TIME)
000018       016          RETURN
000019       016          END

END ELT.
```

C.3 Subroutine M21ADO*STAT01.BOX

```
@ELT,L S.BOX
ELT007 SL73R1 11/01/78 09:35:19 (0,)
000001       000          SUBROUTINE BOX(IX1,IY1,IX2,IY2)
000002       000    C ROD ALLEN, COMP-AID INC., OCTOBER 1978
000003       000    C DRAW A BOX ON TEKTRONIX GIVEN LOWER LEFT AND UPPER RIGHT
000004       000    C CORNERS IN TEKTRONIX SCREEN COORDINATES
000005       000          CALL MOVABS(IX1,IY1)
000006       000          CALL DRWABS(IX2,IY1)
000007       000          CALL DRWABS(IX2,IY2)
000008       000          CALL DRWABS(IX1,IY2)
000009       000          CALL DRWABS(IX1,IY1)
000010       000          RETURN
000011       000          END

END ELT.
```

C.4 Data Element M21ADO*STAT01.DISTAN

```
@ELT,L S1.DISTAN
ELT007 SL73R1 11/01/78 09:38:16 (0,)
000001       000      0.0 0.45 1.15
000002       000     0.45  0.0 0.70
000003       000     1.15 0.70  0.0

END ELT.
```

C.4 Data Element M21ADO*STAT01.ERRLOS

```
@ELT,L S1.ERRLOS
ELT007 SL73R1 11/01/78 09:38:18 (0,)
C00001         000       0.0  1.0  2.0  3.0  4.0
C00002         000       1.0  0.0  1.0  2.0  3.0
C00003         000       2.0  1.0  0.0  1.0  2.0
C00004         000       3.0  2.0  1.0  0.0  1.0
C00005         000       4.0  3.0  2.0  1.0  0.0

END ELT.
```

APPENDIX D

RESIDUAL TIME SERIES LISTINGS

D.1 Control Element Example, M21ADO*STAT02.CASE1

```
@ELT,L S2.CASE1
ELT007 SL73R1 11/01/78 09:39:08 (4,)
000001       001   @XQT M21ADO*STAT02.TIMSER
000002       001   (CHI)(U/G)
000003       003    1 0011
000004       002   01
000005       002   08 TIME LAG
000006       002   15 AUTOCORRELATION
000007       004   01
000008       002   13 FREQUENCY (F)
000009       002   42 VALUE OF NORMALIZED CUMULATIVE PERIODOGRAM
000010       000      101     3  741012  0900  741120  0400
000011       001   @ADD M21ADO*STAT01.COQS2N

END ELT.
```

D.2 Program Map, M21ADO*STAT02.MAPIT

```
@ELT,L S2.MAPIT
ELT007 SL73R1 11/01/78 09:39:23 (3,)
000001      002   @MAP ,M21ADO*STAT02.TIMSER
000002      002   IN M21ADO*STAT02.TIMSER
000003      001   IN M21ADO*STAT02.SETUP
000004      001   IN M21ADO*STAT02.USESET
000005      003   IN SRI*SRI.BOX
000006      003   IN SRI*SRI.HEADER
000007      003   IN RAPS*UTILITY.COMPOZ
000008      000   LIB GRAPH*TEKTRONIX
000009      000   END

END ELT.
```

D.3 Listing of the Main Program, M21ADO*STAT02.TIMSER

```
@ELT,L S2.TIMSER
ELTC07 SL73R1 11/01/78 09:39:35 (12,)
       011   C     *****************************************************************
       011   C
       011   C          THIS PROGRAM COPPUTES THE AUTOCORRELATION AND NORMALIZED CUM-
       011   C     ULATIVE PERIODOGRAM FOR THE RESIDUAL TIME SERIES.
       011   C
       011   C     REFERENCES:
       011   C
       011   C          1.  BOX, G.E. AND G.M. JENKINS, 1976: TIME SERIES ANALYSIS
       011   C          FORECASTING AND CONTROL, HOLDEN-DAY, SAN FRANCISCO, CALIF.
       011   C
       011   C          2.  JENKINS, G.M. AND D.G. WATTS, 12968: SPECTRAL ANALYSIS AND ITS
       011   C          APPLICATIONS, HOLDEN-DAY, SAN FRANCISCO, CALIF.
       011   C
       011   C        *****************************************************************
       011   C
       008   C MODIFIED BY ROD ALLEN, COMP-AID INC., OCTOBER 1978
       011   C
       008         DIMENSION OC(123,3),PC(123,3),IDATE(123),ITIME(123),RTS(123),ACF
       008        1(15),CNP(123),FI(123),XX(2),YY(2),IDEN(5),TIT1(10),TIT2(10),XARR1
       008        2(10),XARR2(10),YARR1(10),YARR2(10)
       009         COMMON /DAT/ XMIN,XMAX,YMIN,YMAX,ITYPE,ICODE,IS
       008         DATA PI,ITYPE /3.1416,0011/
       008     1   FORMAT (//42H ANALYSES OF THE RESIDUAL TIME SERIES FOR ,5A6/)
       008     2   FORMAT (5A6)
       008     3   FORMAT ()
       008     4   FORMAT (/13H NO. OF DATA:,I5,5X,16HNO. OF STATIONS:,I5,5X,6HBDATE:
       008        1I8,5X,6HBTIME:,I6,5X,6HEDATE:,I8,5X,6HETIME:,I6/)
       008     5   FORMAT (/19H NO. OF GOOD DATA =,I5/)
       008     6   FORMAT (/6H SITE:,I3,5X,5HRBAR:,F10.2,5X,6HSUMR2:,F10.2/)
       008     7   FORMAT (/7H A(K) =,15F8.2)
       008     8   FORMAT(I2,1X,10A6)
       008         READ 2,IDEN
       008   CR    PRINT 1 ,IDEN
       008         READ 3,ICODE,ITYPE
       008         READ 8,NTC1,TIT1
       008         READ 8,NXC1,XARR1
       008         READ 8,NYC1,YARR1
       008         READ 8,NTC2,TIT2
       008         READ 8,NXC2,XARR2
       008         READ 8,NYC2,YARR2
       008   C
       008   C     INPUT FROM DATA BASE FILE ...
       008   C
       008   C          ND     = NO. OF DATA
       008   C          NS     = NO. OF STATIONS.
       008   C          IDATEB = BEGIN DATE OF INTEREST.
       008   C          ITIMEB = BEGIN TIME OF INTEREST.
       008   C          IDATEE = END DATE OF INTEREST.
       008   C          ITIMEE = END TIME OF INTEREST.
       008   C          IDATES = DATE OF OBSERVATION
       008   C          ITIMES = TIME OF OBSERVATION
       008   C          ISITE = STATION NO.
       008   C          ISTC,IWSC,IWDC ARE NOT USED.
       008   C          OCS    = OBSERVED CONCENTRATION
       008   C          PCS    = PREDICTED CONCENTRATION
```

```
000056    008   C
000057    008   C
000058    008   C
000059    008   C      INPUT FROM PARAMETER FILE ...
000060    008   C
000061    008   C            IDEN = DATA IDENTIFIER
000062    008   C            ICODE = 1, AUTOCORRELATION ANALYSIS.
000063    008   C                  = 2, NORMALIZED CUMULATIVE PERIODOGRAM
000064    008   C                  = 3, DO CODES 1 AND 2.
000065    008   C            ITYPE = 0001, X-AXIS IS LINEAR.
000066    008   C                  = 0010, Y-AXIS IS LINEAR.
000067    008   C                  = 0100, X-AXIS IS LOGARITHMIC.
000068    008   C                  = 1000, Y-AXIS IS LOGARITHMIC.
000069    008   C            NTC1  = NO. OF ASCII CHARACTERS IN TIT1.
000070    008   C            TIT1  = UPPER TITLE FOR GRAPH 1.
000071    008   C            NXC1  = NO. OF ASCII CHARACTERS IN XARR1.
000072    008   C            XARR1 = X-AXIS LABEL FOR GRAPH 1.
000073    008   C            NYC1  = NO. OF ASCII CHARACTERS IN YARR1.
000074    008   C            YARR1 = Y-AXIS LABEL FOR GRAPH 1.
000075    008   C            NTC2  = NO. OF ASCII CHARACTERS IN TIT2.
000076    008   C            TIT2  = UPPER TITLE FOR GRAPH 2.
000077    008   C            NXC2  = NO. OF ASCII CHARACTERS IN XARR2.
000078    008   C            XARR2 = X-AXIS LABEL FOR GRAPH 2.
000079    008   C            NYC2  = NO. OF ASCII CHARACTERS IN YARR2.
000080    008   C            YARR2 = Y-AXIS LABEL FOR GRAPH 2.
000081    008   C
000082    008          READ 3,ND,NS,IDATEB,ITIMEB,IDATEE,ITIMEE
000083    008   CR     PRINT4,ND,NS,IDATEB,ITIMEB,IDATEE,ITIMEE
000084    008          DO 110 I=1,NS
000085    008          DO 100 J=1,ND
000086    008          OC(J,I)=-99.9
000087    008          PC(J,I)=-99.9
000088    008     100  CONTINUE
000089    008     110  CONTINUE
000090    008          NT=1
000091    008          DO 135 I=1,ND
000092    008          READ 3,IDATES,ITIMES,ISITE,ISTC,IWSC,IWDC,OCS,PCS
000093    008          IF (IDATES.EQ.IDATEB.AND.ITIMES.LT.ITIMEB) GO TO 135
000094    008          IF (IDATES.EQ.IDATEE.AND.ITIMES.GT.ITIMEE) GO TO 135
000095    008          IF (I.GT.1) GO TO 115
000096    008          IDATED=IDATES
000097    008          ITIMED=ITIMES
000098    008     115  IF (IDATES.EQ.IDATED) GO TO 120
000099    008          IDATED=IDATES
000100    008          NT=NT+1
000101    008          IDATE (NT)=IDATED
000102    008          GO TO 125
000103    008     120  IF (ITIMES.EQ.ITIMED) GO TO 130
000104    008          NT=NT+1
000105    008     125  ITIMED=ITIMES
000106    008          ITIME (NT)=ITIMED
000107    008     130  OC(NT,ISITE)=OCS
000108    008          PC(NT,ISITE)=PCS
000109    008     135  CONTINUE
000110    008   CR     PRINT 5,NT
000111    008          PI2=2.0*PI
000112    008          RNT=NT
```

```
008              ROOTN=SQRT(RNT)
008              TWON=2.0/NT
008              JN=NT/2
012              RJN=JN
008       CR     PRINT 10,RNT,ROOTN,TWON,PI2
012       C10    FORMAT (/1H ,10F12.4)
008              DO 300 IS=1,NS
008       C            DETERMINE THE RESIDUAL TIME SERIES.
008              SUMR=0.0
008              DO 140 I=1,NT
008              OCS=OC(I,IS)
008              PCS=PC(I,IS)
008              IF (OCS.LT.0.0.OR.PCS.LT.0.0) GO TO 139
008              RTS(I)=OCS-PCS
008              SUMR=SUMR+RTS(I)
008              GO TO 140
012       C              WHENEVER OC OR PC IS INVALID SET RTS = -99999.9
012        139   RTS(I)=-99999.9
008        140   CONTINUE
008              RBAR=SUMR/NT
011       C           DETERMINE THE AUTOCORRELATION FUNCTION (ACF) OF THE R.T.S.
011       C             RTS = RESIDUALS
011       C             RBAR = MEAN OF THE RESIDUALS
011       C             SUMR2 = VARIANCE OF THE RESIDUALS
011       C             NTMK = TIME LAG
011       C             NT = NO. OF POINTS
011       C
008              SUMR2=0.0
008              DO 145 I=1,NT
008              RMR=RTS(I)-RBAR
008              SUMR2=SUMR2+RMR*RMR
008        145   CONTINUE
008       CR     PRINT 6,IS,RBAR,SUMR2
012              IF (ICODE-2) 150,190,150
008        150   XMIN=0.0
008              XMAX=16.0
008              YMIN=-1.0
008              YMAX= 1.0
008              DO 160 K=1,JN
008              SUM=0.0
008              NTMK=NT-K
008              DO 155 L=1,NTMK
012              IF (RTS(L).LE.-99999.9.OR.RTS(L+K).LE.-99999.9) GO TO 155
008              SUM=SUM+(RTS(L)-RBAR)*(RTS(L+K)-RBAR)
008        155   CONTINUE
008       CR     ACF(K)=SUM/SUMR2
008              ACF(K)=(SUM/SUMR2)/(NTMK/RNT)
008        160   CONTINUE
008       CR     PRINT 7,(ACF(K),K=1,JN)
008       C           PLOT THE AUTOCORRELATION FUNCTION AND CONFIDENCE BANDS.
012       C
012       C             IN THE FOLLOWING DO-LOOP THE UPPER LIMIT OF K DETERMINES
012       C      THE MAXIMUM TIME LAG.  AS A DEFAULT IT CAN BE SET TO HALF THE
012       C      NUMBER OF SAMPLES (JN) IN THE TIME SERIES.
012       C
008              AUL=1.96/ROOTN
008              ALL=-AUL
```

```
008           CALL SETUP (NTC1,TIT1,NXC1,XARR1,NYC1,YARR1)
008   C             DRAW THE PLUS AND MINUS 95 PERCENT CONFIDENCE LIMITS
008           XX(1)=0.0
008           YY(1)=AUL
008           XX(2)=XMAX
008           YY(2)=AUL
008           CALL LINE (12)
009           CALL XNEAT(0)
009           CALL XTICS(2)
008           CALL NPTS (2)
008           CALL CHECK (XX,YY)
008           CALL DSPLAY (XX,YY)
008           YY(1)=ALL
008           YY(2)=ALL
008           CALL CPLOT (XX,YY)
008   C             DRAW THE ZERO LINE.
008           XX(1)=0.0
008           XX(2)=XMAX
008           YY(1)=0.0
008           YY(2)=0.0
008           CALL LINE (0)
008           CALL NPTS (2)
008           CALL CPLOT (XX,YY)
008   C             DRAW AUTOCORRELATION VALUES AT EACH K-TIME PERIOD.
008           DO 165 K=1,15
008   C RHA 10/4/78 C /AFC/ACF/
008           YC=ACF(K)
008           XC=K
008           CALL MOVEA (XC,0.0)
008           CALL DRAWA (XC,YC)
008      165  CONTINUE
008           CALL FRAME
008           CALL HDCOPY
008           IF (ICODE.EQ.1) GO TO 300
008   C             DETERMINE THE NORMALIZED CUMMULATIVE PERIODOGRAM.
011   C             CNP = PARTIAL SUM OF I(FI)/SUM OF I(FI)
011   C             FI = FREQUENCY
011   C             TWON = 2.0/NT
011   C
012      190  J=0
008      200  J=J+1
008           FI(J)=J/RNT
008           IF (J.GT.JN) GO TO 230
008           SUM=0.0
008           DO 220 I=1,J
008           FIPI2=FI(I)*PI2
008           SUMC=0.0
008           SUMS=0.0
008           DO 210 L=1,NT
012           IF (RTS(L).LE.-99999.9) GO TO 210
008           FIPL=FIPI2*L
008           SUMC=SUMC+RTS(L)*COS(FIPL)
008           SUMS=SUMS+RTS(L)*SIN(FIPL)
008      210  CONTINUE
008           FIF=TWON*(SUMC*SUMC+SUMS*SUMS)
008           SUM=SUM+FIF
008      220  CONTINUE
```

```
008             CNP(J)=SUM
008             GO TO 200
008   C               DETERMINE THE STRAIGHT LINE PARAMETERS.
008        230  DO 240 J=1,JN
008             CNP(J)=CNP(J)/CNP(JN)
008        240  CONTINUE
008             XMIN=0.0
008             YMIN=0.0
008             XMAX=0.5
008             YMAX=1.0
008             DLINE=1.96/ROOTN
008             SLOPE=YMAX/XMAX
008             ANGLE=ATAN(SLOPE)
008             COSA=COS(ANGLE)
008             B=DLINE/COSA
008   CR        PRINT 10,DLINE,SLOPE,ANGLE,COSA,B
008   C               PLOT THE NORMALIZED CUMMULATIVE PERIODOGRAM VS. FREQUENCY.
008             CALL SETUP (NTC2,TIT2,NXC2,XARR2,NYC2,YARR2)
008             CALL STEPS(1)
009   CR        CALL SIZES (0.5)
010             CALL LINE(-1)
008             CALL SYMBL (1)
008             CALL NPTS (JN)
008             CALL CHECK (FI,CNP)
008             CALL DSPLAY (FI,CNP)
008             CALL LINE(0)
009             CALL SYMBL(0)
008             CALL NPTS(2)
009   C               LINE THRU (0.0,0.0) AND (0.5,1.0)
009             XX(1)=XMIN
009             YY(1)=YMIN
009             XX(2)=XMAX
009             YY(2)=YMAX
008             CALL CPLOT(XX,YY)
008   C
008   C               PLOT 95 PERCENT CONFIDENCE BANDS Y=MX+B WHERE M=2 AND B IS TO
008   C         BE DETERMINED.
008   C
008   C               PLOT UPPER 95 PERCENT CONFIDENCE BAND.
008             XX(1)=XMIN
008             YY(1)=B
009             XX(2)=XMAX-B/SLOPE
008             YY(2)=YMAX
008             CALL LINE (12)
008             CALL NPTS (2)
008             CALL CPLOT (XX,YY)
008   C               PLOT THE LOWER 95 PERCENT CONFIDENCE BAND.
008             XX(1)=B/SLOPE
008             YY(1)=YMIN
008             XX(2)=XMAX
009             YY(2)=YMAX-B
008             CALL CPLOT (XX,YY)
008   C               PLACE THE SECOND AXIS AT THE TOP OF THE BOX.
009   CR        CALL DINITX
009   CR        CALL XLOCTP (0)
009   CR        CALL XTICS (8)
009   CR        CALL XLAB (-1)
```

```
C00284        009   CR     CALL CHECK (XX,YY)
C00285        009   CR     CALL DSPLAY (XX,YY)
C00286        008          CALL FRAME
C00287        008          CALL HDCOPY
C00288        008     300  CONTINUE
C00289        008          CALL FINITT (0,700)
C00290        008          END
END ELT.
```

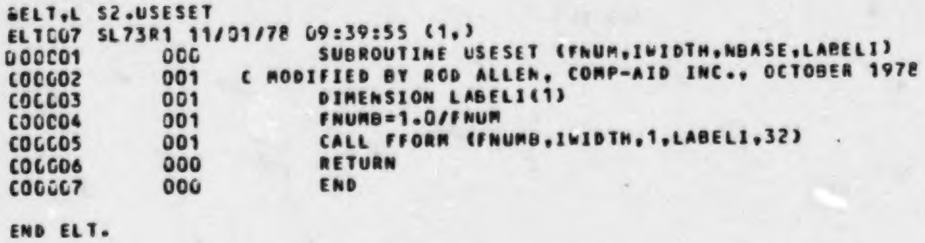
D.4 Subroutine M21ADO*STATO2.SETUP

```
@ELT,L S2.SETUP
ELT007 SL73R1 11/01/78 09:39:31 (8,)
000001       004           SUBROUTINE SETUP (NTC,TIT,NXC,XARR,NYC,YARR)
000002       004    C MODIFIED BY ROD ALLEN, COMP-AID INC., OCTOBER 1978
000003       004    C
000004       004    C            THIS ROUTINE SETS UP THE TEKTRONIX GRAPHICS PACKAGE, SETS THE
000005       004    C      INITIAL LIMITS FOR THE GRAPH AND DRAWS THE FRAME AND LABELS.
000006       004    C
000007       006           COMMON /DAT/ XMIN,XMAX,YMIN,YMAX,ITYPE,ICODE,IS
000008       004           DIMENSION XARR(1),YARR(1),TIT(1)
000009       004           CALL INITT (120)
000010       004           CALL TERM(2,1024)
000011       004    C ASSUMES TEKTRONIX 4014
000012       004           CALL HEADER
000013       004           CALL CHRSIZ(2)
000014       004           CALL BOX(0,0,1023,780)
000015       006           CALL MOVABS(10,760)
000016       006           CALL AOUTST(17,'TIME SERIES, CASE')
000017       006           ENCODE(2,98,CASE) ICODE
000018       007           CALL AOUTST(2,CASE)
000019       006           CALL AOUTST(10,', STATION ')
000020       007           ENCODE(2,98,CASE) IS
000021       006    98     FORMAT(I2)
000022       006           CALL AOUTST(2,CASE)
000023       004           CALL BINITT
000024       004    C            SET THE SCREEN LIMITS FOR X AND Y.
000025       004           CALL PLACE (3HSTD)
000026       004           CALL XFRM (3)
000027       004           CALL YFRM (3)
000028       004           CALL XMFRM (3)
000029       004           CALL YMFRM (3)
000030       004    C            PLOT THE MAJOR-MINOR TIC MARKS ON THE AXES.
000031       004           CALL DLIMX (XMIN,XMAX)
000032       004           CALL DLIMY (YMIN,YMAX)
000033       004           CALL XNEAT (1)
000034       004           CALL YNEAT (1)
000035       004    C            DETERMINE THE (X,Y) AXES TYPE.
000036       004           IF (ITYPE.GT.100) CALL YTYPE(2)
000037       004           ITYPE=ITYPE-1000
000038       004           IF (ITYPE.GT.10) CALL XTYPE(2)
000039       004           ITYPE=ITYPE-100
000040       004           IF (ITYPE.GT.1) CALL YTYPE(1)
000041       004           ITYPE=ITYPE-10
000042       004           IF (ITYPE.GT.0) CALL XTYPE(1)
000043       004           CALL MOVABS(300,50)
000044       004           CALL AOUTST(NXC,XARR)
000045       004           CALL QUOTE(50.,150.,.15,YARR,90.,NYC)
000046       004           CALL MOVABS(300,30)
000047       004           CALL AOUTST(NTC,TIT)
000048       004           IF (ICODE.LE.1) GO TO 100
000049       004           CALL MOVABS(300,630)
000050       004           CALL AOUTST(34,'DASHED LINES 95% CONFIDENCE LIMITS')
000051       004           IF (XMAX.GT.1.0) GO TO 100
000052       008    CR     CALL MOVABS(300,760)
000053       008    CR     CALL AOUTST(12,'PERIOD (1/F)')
000054       004    C            DRAW A STANDARD FRAME.
000055       004     100   CALL FRAME
```

```
000056        004         RETURN
000057        004         END

END ELT.
```

D.4 Subroutine M21ADO*STATO2.USESET



```
@ELT,L S2.USESET
ELT007 SL73R1 11/01/78 09:39:55 (1,)
000001       000          SUBROUTINE USESET (FNUM,IWIDTH,NBASE,LABELI)
000002       001    C MODIFIED BY ROD ALLEN, COMP-AID INC., OCTOBER 1978
000003       001          DIMENSION LABELI(1)
000004       001          FNUMB=1.0/FNUM
000005       001          CALL FFORM (FNUMB,IWIDTH,1,LABELI,32)
000006       000          RETURN
000007       000          END

END ELT.
```

APPENDIX E

CHI-SQUARE GOODNESS-OF-FIT LISTINGS

E.1 Control Element Example, M21ADO*STAT03.CASE1

```
@ELT,L S3.CASE1
 ELT007 SL73R1 11/01/78 09:40:46 (4,)
C00001        001    @XQT M21ADO*STAT03.CHIFIT
C00002        000    (CHI)(U)/(G)
C00003        004    1 0011
C00004        003    21 OBSERVED DISTRIBUTION
C00005        003    22 PREDICTED DISTRIBUTION
C00006        003    18 OBSERVED-PREDICTED
C00007        003    18 OBSERVED-PREDICTED
C00008        003    06 NUMBER
C00009        003    06 NUMBER
C00010        003    06 NUMBER
C00011        003    07 PERCENT
C00012        000       101      3
C00013        001    @ADD M21ADO*STAT01.COQS2N

END ELT.
```

E.2 Program Map, M21ADO*STATO3.MAPIT

```
@ELT,L S3.MAPIT
ELT007 SL73R1 11/01/78 09:41:38 (2,)
C00001      001   @MAP ,M21ADO*STAT03.CHIFIT
C00002      001   IN M21ADO*STAT03.CHIFIT
C00003      001   IN M21ADO*STAT03.HISTO
C00004      001   IN M21ADO*STAT03.SETUP
C00005      001   IN M21ADO*STAT03.ULINE
C00006      002   IN SRI*SRI.BOX
C00007      002   IN SRI*SRI.HEADER
C00008      002   IN RAPS*UTILITY.COMPO2
C00009      000   LIB GRAPH*TEKTRONIX
C00010      000   END

END ELT.
```

E.3 Listing of the Main Program, M21ADO*STAT03.CHIFIT

```
@ELT,L S3.CHIFIT
ELT007 SL73R1 11/01/78 09:41:17 (8,)
004   C
004   C          CHI-SQUARE GOODNESS-OF-FIT TEST.
004   C
008   C          THIS PROGRAM COMPUTES THE CHI-SGUARE STATISTIC AND PLOTS
008   C       FREQUENCY DISTRIBUTION FOR PREDICTED AND OBSERVED CONCENTRATIONS,
008   C       THE DIFFERENCES AND PERCENTAGES.
006   C
004   C MODIFIED BY ROD ALLEN, COMP-AID INC., OCTOBER 1978
004         COMMON /DAT/ XMIN,XMAX,YMIN,YMAX,NCLAS,XCLAS(21),ICLAS(21),ITYPE
004         COMMON /MNM/ XMIN1,XMIN2,XMAX1,XMAX2,YMIN1,YMIN2,YMAX1,YMAX2
004         DIMENSION IDATE(123),ITIME(123),ISITE(123),OC(123),PC(123),IDEN(5)
004        1,OBSD(21),PRED(21),PMOD(21),TIT1(10),TIT2(10),TIT3(10),TIT4(10),
004        2YARR1(10),YARR2(10),YARR3(10),YARR4(10),PMOP(21)
004         DATA NCLAS,NCLAS1 /10,11/
004    1    FORMAT (//37H CHI SQUARE GOODNESS-OF-FIT TEST FOR ,5A6/)
004    2    FORMAT (10A6)
004    3    FORMAT ()
004    4    FORMAT (/5H NN =,I5,5X,4HNS =,I5/)
004    5    FORMAT (/7H ICODE:,I5,3X,6HITYPE:,I5//)
004    6    FORMAT (/28H NO. OF NEG DATA DISCARDED =,I4/)
004    7    FORMAT(I2,1X,10A6)
004   C RHA ALLOW UP TO 30 CHARACTERS IN TITLES
004   C
004   C     INPUT FROM PARAMETER FILE ...
004   C
004   C          IDENT = DATA IDENTIFIER.
004   C          ICODE = 1, PLOT OC AND PC HISTOGRAMS.
004   C                  2, PLOT OC-PC HISTOGRAMS.
004   C                  3, DO CODES 1 AND 2.
004   C          ITYPE = 0001, X-AXIS IS LINEAR.
004   C                = 0010, Y-AXIS IS LINEAR
004   C                = 0100, X-AXIS IS LOGARITHMIC.
004   C                = 1000, Y-AXIS IS LOGARITHMIC.
004   C          NTC   = NO. OF ASCII CHARACTERS IN THE UPPER TITLE.
004   C          NXC   = NO. OF ASCII CHARACTERS IN THE X-AXIS LAPEL.
004   C          NYC   = NO. OF ASCII CHARACTERS IN THE Y-AXIS LABEL.
004   C          NZC   = NO. OF ASCII CHARACTERS IN THE LOWER TITLE.
004   C          TIT   = GRAPH UPPER TITLE.
004   C          XARR  = X-AXIS LABEL.
004   C          YARR  = Y-AXIS LABEL.
004   C          ZARR  = GRAPH LOWER TITLE.
004   C
004         READ 2,IDEN
004   CR    PRINT 1,IDEN
004         READ 3,ICODE,ITYPE
004   CR    PRINT5,ICODE,ITYPE
004         READ 7,NTC1,TIT1
004         READ 7,NTC2,TIT2
004         READ 7,NTC3,TIT3
004         READ 7,NTC4,TIT4
004         READ 7,NYC1,YARR1
004         READ 7,NYC2,YARR2
004         READ 7,NYC3,YARR3
004         READ 7,NYC4,YARR4
004   C
```

```
004   C     *INPUT FROM DATA BASE FILE ...()
004   C
004   C           ND = NO. OF DATA.
004   C           NS = NO. OF STATIONS.
004   C           IDATE = DATE OF OBSERVATION
004   C           ISITE = STATION NO.
004   C           ITIME = TIME OF OBSERVATION
004   C           ISTC,IWSC,IWDC ARE NOT USED.
004   C           OC = OBSERVED CONCENTRATION
004   C           PC = PREDICTED CONCENTRATION
004   C
004         READ 3,ND,NS
004   CR    PRINT 4,ND,NS
004         N=0
004         XMIN1=+1.0E10
004         XMAX1=-1.0E10
004         DO 60 I=1,ND
004         READ 3,IDATE(I),ITIME(I),ISITE(I),ISTC,IWSC,IWDC,OC(I),PC(I)
006   CR    PRINT 99, IDATE(I),ITIME(I)
005   99    FORMAT(2I10)
004         IF (OC(I).LT.0.0.OR.PC(I).LT.0.0) GO TO 50
004         N=N+1
004         OC(N)=OC(I)
004         PC(N)=PC(I)
004         XMAX1=AMAX1(XMAX1,PC(I),OC(I))
004         XMIN1=AMIN1(XMIN1,PC(I),OC(I))
004         GO TO 60
004    50   NEG=NEG+1
006   CR    PRINT 6,NEG
004    60   CONTINUE
004   C
004   C          DETERMINE THE NUMBER OF OC IN EACH OF 10 CLASSES.
004   C
004         DELTX=(XMAX1-XMIN1)/NCLAS
004         ITEN=DELTX*0.1
004         TENS=ITEN*10.0
004         REM=DELTX-TENS
004         IF (REM.GT.0.0) DELTX=TENS+10.0
004         ICLAS(1)=0
004         DO 100 I=2,NCLAS1
004         XCLAS(1)=0
004         ICLAS(I)=ICLAS(I-1)+DELTX
004         XCLAS(I)=ICLAS(I)
004    100  CONTINUE
004         DO 110 I=1,N
004         IND=OC(I)/DELTX+1.0
004         IF (IND.GT.NCLAS) IND=NCLAS
004         OBSD(IND)=OBSD(IND)+1.0
004    110  CONTINUE
004         OBSD(NCLAS1)=OBSD(NCLAS)
004         YMIN1=0.0
004         YMAX1=-1.0E10
004   C          DETERMINE YMAX FOR ALL CLASSES.
004         DO 115 I=1,NCLAS
004         YMAX1=AMAX1(YMAX1,OBSD(I))
004    115  CONTINUE
004   C
```

```
000113        004    C          DETERMINE THE NUMBER OF PC IN EACH OF 10 CLASSES.
000114        004    C
000115        004             DO 120 I=1,N
000116        004             IND=PC(I)/DELTX+1.0
000117        004             IF (IND.GT.NCLAS) IND=NCLAS
000118        004             PRED(IND)=PRED(IND)+1.0
000119        004       120   CONTINUE
000120        004             PRED(NCLAS1)=PRED(NCLAS)
000121        004             DO 125 I=1,NCLAS
000122        004             YMAX1=AMAX1(YMAX1,PRED(I))
000123        004       125   CONTINUE
000124        004             YMAX2=YMAX1
000125        004             YMIN2=YMIN1
000126        007             CHI2=0.
000127        004             IF (ICODE-2) 130,135,130
000128        004       130   CALL HISTO (NCLAS1,OBSD,PRED,NTC1,NTC2,TIT1,TIT2,NYC1,NYC2,YARR1,
000129        007            1YARR2,ICODE,CHI2,N)
000130        004             IF (ICODE.EQ.1) GO TO 150
000131        004    C
000132        004    C          DETERMINE THE NUMBER OF PC-OC IN EACH OF 10 CLASSES.
000133        004    C
000134        004       135   YMAX1=-1.0E10
000135        004             YMIN1=+1.0E10
000136        004             DO 140 I=1,NCLAS1
000137        004             PMOD(I)=PRED(I)-OBSD(I)
000138        004             YMAX1=AMAX1(YMAX1,PMOD(I))
000139        004             YMIN1=AMIN1(YMIN1,PMOD(I))
000140        004       140   CONTINUE
000141        004    C
000142        004    C          DETERMINE THE PERCENTAGES OF (PC-OC)/OC IN EACH OF 10 CLASSES.
000143        004    C
000144        004             YMAX2=-1.0E10
000145        004             YMIN2=+1.0E10
000146        004             DO 145 I=1,NCLAS1
000147        007             IF(OBSD(I).LE.1.) OBSD(I)=1.
000148        007             CHI2=CHI2+PMOD(I)*PMOD(I)/OBSD(I)
000149        004             PMOP(I)=100.0*PMOD(I)/OBSD(I)
000150        004             YMAX2=AMAX1(YMAX2,PMOP(I))
000151        004             YMIN2=AMIN1(YMIN2,PMOP(I))
000152        004       145   CONTINUE
000153        004             CALL HISTO (NCLAS1,PMOD,PMOP,NTC3,NTC4,TIT3,TIT4,NYC3,NYC4,YARR3,
000154        007            1YARR4,ICODE,CHI2,N)
000155        004       150   CALL FINITT (0,700)
000156        004             STOP
000157        004             END

END ELT.
```

E.4 Subroutine M21ADO*STAT03.HISTO

```
@ELT,L S3.HISTO
ELT007 SL73R1 11/01/78 09:41:30 (13,)
C00001        009           SUBROUTINE HISTO (NC1,Y1,Y2,NTC1,NTC2,TIT1,TIT2,NYC1,NYC2,YARR1,
C00002        012          1YARR2,ICODE,CHI2,N)
C00003        009     C MODIFIED BY ROD ALLEN, COMP-AID INC., OCTOBER 1978
C00004        009           COMMON /DAT/ XMIN,XMAX,YMIN,YMAX,NCLAS,XCLAS(21),ICLAS(21),ITYPE
C00005        009           DIMENSION Y1(NC1),TIT1(1),YARR1(1),Y2(NC1),TIT2(1),
C00006        013          1YARR2(1),YY(21),AOUT(6)
C00007        009           COMMON /MNM/XMIN1,XMIN2,XMAX1,XMAX2,YMIN1,YMIN2,YMAX1,YMAX2
C00008        009           DATA YY /21*0.0/
C00009        009     CR 1  FORMAT (/7H CLASS:,10(2XI4,1H-,I4))
C00010        009     CR 2  FORMAT (1H ,10(2XF9.0))
C00011        009     CR    PRINT 1,(ICLAS(J),ICLAS(J+1),J=1,10)
C00012        009     CR    PRINT 2,(Y1(J),J=1,10)
C00013        009           IF (NCLAS.LE.10) GO TO 100
C00014        009     CR    PRINT 1,(ICLAS(J),ICLAS(J+1),J=11,NCLAS)
C00015        009     CR    PRINT 2,(Y1(J),J=11,NC1)
C00016        009     100   CONTINUE
C00017        009     CR    PRINT 1,(ICLAS(J),ICLAS(J+1),J=1,10)
C00018        009     CR    PRINT 2,(Y2(J),J=1,10)
C00019        009           IF (NCLAS.LE.10) GO TO 110
C00020        009     CR    PRINT 1,(ICLAS(J),ICLAS(J+1),J=11,NCLAS)
C00021        009     CR    PRINT 2,(Y2(J),J=11,NC1)
C00022        009      110  CALL INITT (120)
C00023        009     C RHA ASSUMES TEKTRONIX 4014
C00024        009           CALL TERM(2,1024)
C00025        009     C RHA PRINT DATE AND TIME HEADER
C00026        009           CALL HEADER
C00027        009           XMIN=XMIN1
C00028        009           XMAX=XMAX1
C00029        009           YMIN=YMIN1
C00030        009           YMAX=YMAX1
C00031        009           LABYY=540
C00032        009           CALL SETUP (NTC1,TIT1,NYC1,YARR1,3HUPH,450,LABYY,730)
C00033        009     C             PLOT UPPER HISTOGRAMS.
C00034        010     CR ORIGINAL AGII USED -11 FOR USER DEFINE LINE
C00035        010           CALL LINE (-11)
C00036        009           CALL NPTS (NC1)
C00037        009           CALL CHECK (XCLAS,Y1)
C00038        009           CALL DSPLAY (XCLAS,Y1)
C00039        009     C             PLOT DASHED ZERO LINE.
C00040        009           CALL LINE (12)
C00041        009           CALL CPLOT (XCLAS,YY)
C00042        009           CALL FRAME
C00043        009           YMIN=YMIN2
C00044        009           YMAX=YMAX2
C00045        009           LABYY=150
C00046        009           CALL SETUP (NTC2,TIT2,NYC2,YARR2,3HLOH,75,LABYY,350)
C00047        009     C             PLOT LOWER HISTOGRAMS.
C00048        010           CALL LINE (-11)
C00049        009           CALL NPTS (NC1)
C00050        009           CALL CHECK (XCLAS,Y2)
C00051        009           CALL DSPLAY (XCLAS,Y2)
C00052        009     C             PLOT DASHED ZERO LINE.
C00053        009           CALL LINE (12)
C00054        009           CALL CPLOT (XCLAS.YY)
C00055        009           CALL FRAME
```

```
000056       011             CALL MOVABS (10,760)
000057       009             CALL AOUTST(33,"CHI-SQUARE GOODNESS-OF-FIT, CASE ")
000058       009             ENCODE(1,101,CASE) ICODE
000059       009     101     FORMAT(I1)
000060       009             CALL AOUTST(1,CASE)
000061       009     C RHA BOX AROUND PLOT
000062       009             CALL BOX(0,0,1023,780)
000063       009             CALL BOX(0,0,1023,390)
000064       012     C PRINT CHI2 AND N IN BOX
000065       012             IF(CHI2.LE.0.) GO TO 80
000066       012             CALL BOX(849,0,1023,33)
000067       013             ENCODE(36,81,AOUT) CHI2,N
000068       012     81      FORMAT("CHI-SQUARE =",1PG8.2,4X,"N =",I6,3X)
000069       012             CALL MOVABS(853,4)
000070       012             CALL CHRSIZ(3)
000071       013             CALL AOUTST(9,AOUT(5))
000072       012             CALL MOVABS(853,18)
000073       012             CALL AOUTST(20,AOUT)
000074       012             CALL CHRSIZ(2)
000075       012     80      CALL HDCOPY
000076       009             RETURN
000077       012             END

END ELT.
```

E.4 Subroutine M21ADO*STAT03.SETUP

```
@ELT,L S3.SETUP
ELT007 SL73R1 11/01/78 09:41:45 (6,)
000001       002         SUBROUTINE SETUP (NTC,TIT,NYC,YARR,IPLOT,LABXY,LABYY,LABTY)
000002       002   C MODIFIED BY ROD ALLEN, COMP-AID INC., OCTOBER 1978
000003       002   C
000004       002   C            THIS ROUTINE SETS UP THE TEKTRONIX GRAPHICS PACKAGE, SETS THE
000005       002   C     INITIAL LIMITS FOR THE GRAPH AND DRAWS THE FRAME AND LABELS.
000006       002   C
000007       002         DIMENSION TIT(1),YARR(1)
000008       002         COMMON /DAT/ XMIN,XMAX,YMIN,YMAX,NCLAS,XCLAS(21),ICLAS(21),ITYPE
000009       002         CALL BINITT
000010       006         CALL CHRSIZ(2)
000011       002   C            SET THE SCREEN LIMITS FOR X AND Y.
000012       002         CALL PLACE (IPLOT)
000013       002         CALL XFRM (3)
000014       002         CALL YFRM (3)
000015       002         CALL XMFRM (3)
000016       002         CALL YMFRM (3)
000017       002   C            PLOT THE MAJOR-MINOR TIC MARKS ON THE AXES.
000018       002         CALL DLIMX (XMIN,XMAX)
000019       002         CALL DLIMY (YMIN,YMAX)
000020       002         CALL XNEAT (1)
000021       002         CALL YNEAT (1)
000022       002   C            DETERMINE THE (X,Y) AXES TYPE.
000023       002         IF (ITYPE.GT.100) CALL YTYPE(2)
000024       002         ITYPE=ITYPE-1000
000025       002         IF (ITYPE.GT.10) CALL XTYPE(2)
000026       002         ITYPE=ITYPE-100
000027       002         IF (ITYPE.GT.1) CALL YTYPE(1)
000028       002         ITYPE=ITYPE-10
000029       002         IF (ITYPE.GT.0) CALL XTYPE(1)
000030       003         CALL MOVABS(300,LABXY)
000031       002         CALL AOUTST(19,"CONCENTRATION CLASS")
000032       002   C RHA QUOTE IS EPA NCC CALCOMP SYMBOL ROUTINE FOR TEKTRONIX
000033       004         YPAGE=LABYY
000034       004         CALL QUOTE(50.,YPAGE,.15,YARR,90.,NYC)
000035       005         CALL TSEND
000036       002         CALL MOVABS(300,LABTY)
000037       002         CALL AOUTST(NTC,TIT)
000038       002         RETURN
000039       002         END

END ELT.
```

APPENDIX F

BIVARIATE REGRESSION AND CORRELATION LISTING

F.1 Control Element Example, M21ADO*STAT04.CASE6

```
@ELT,L S4.CASE6
ELT007 SL73R1 11/01/78 09:43:42 (4,)
C00001        001    @XQT M21ADO*STAT04.REGANA
C00002        000    (CHI)(U)/(Q)
C00003        004     6 0011
C00004        000    6.31 2.92 2.35 2.13 2.02 1.94 1.90 1.86 1.83 1.81 1.80 1.78 1.77 1.76 1.75 1.75
C00005        000    1.74 1.73 1.73 1.72 1.72 1.72 1.71 1.71 1.71 1.71 1.70 1.70 1.70 1.70 1.68 1.67
C00006        000    1.66 1.645
C00007        003    36 (CHI)(U/Q) -- UNITS (10**-6)*(M**-2)
C00008        003    22 OBSERVED CONCENTRATION
C00009        003    23 PREDICTED CONCENTRATION
C00010        003    11 SCATTERGRAM
C00011        003    28 REGRESSION LINE (SOLID LINE)
C00012        003    30 SENSITIVITY BANDS (SMALL DASH)
C00013        003    33 90% CONFIDENCE BANDS (SMALL DASH)
C00014        003    30 PROBABILITY BANDS (LARGE DASH)
C00015        003    58 90% CONFIDENCE (SMALL DASH) PROBABILITY (LARGE DASH) BANDS
C00016        002     101 3
C00017        001    @ADD M21ADO*STAT01.COBS2N

END ELT.
```

F.2 Program Map, M21ADO*STAT04.MAPIT

```
@ELT,L S4.MAPIT
ELT007 SL73R1 11/01/78 09:44:00 (2,)
000001       001   @MAP ,M21ADO*STAT04.REGANA
000002       001   IN M21ADO*STAT04.REGANA
000003       001   IN M21ADO*STAT04.STAB
000004       001   IN M21ADO*STAT04.LINLSQ
000005       002   IN SRI*SRI.BOX
000006       002   IN SRI*SRI.HEADER
000007       002   IN RAPS*UTILITY.COMPOZ
000008       000   LIB GRAPH*TEKTRONIX
000009       000   END

END ELT.
```

F.3 Listing of the Main Program, M21ADO*STATO4.REGANA

```
@ELT,L S4.REGANA
ELT007 SL73R1 11/01/78 09:44:08 (15,)
000001       011    C REGRESSION ANALYSIS PROGRAM
000002       011    C MODIFIED BY ROD ALLEN, COMP-AID INC., OCTOBER 1978
000003       011          COMMON /LSG/ RECNOP,A,B,XSD,YSD,CIL,CIU,SXMXB,RC,RMSE,ESD,XBAR
000004       011          DIMENSION X1(39),X2(39),X3(39),Y1(39),Y2(39),Y3(39),YRL(123),IDATE
000005       011         1(123),ITIME(123),ISITE(123),OC(123),PC(123),IDEN(5),TAB(34),XARR
000006       011         2(10),YARR(10),TIT(10),XX(123),YY(123),ZARR(10),ZAR(10),AOUT(8)
000007       011          DIMENSION YCLP(123),YCLM(123),YPLP(123),YPLM(123),XS(123),YS(123)
000008       011          DATA XMIN,YMIN,XMAX,YMAX /10.,10.,-1.0E10,-1.0E10/
000009       011          DATA N1,N2,N3,NEG,YBIG /0,0,0,0,-1.0E10/
000010       011     1    FORMAT (//49H STATISTICAL ANALYSIS FOR OBSERVED VS. PREDICTED 5A6)
000011       011     2    FORMAT (5A6)
000012       011     3    FORMAT ()
000013       011     4    FORMAT (/5H ND =,I5,5X,4HNS =,I5/)
000014       011     5    FORMAT (/7H ICODE:I3,3X6HITYPE:I5,3X4HNTC:,I3,3X4HNXC:I3,3X4HNYC:
000015       011         1I3,/)
000016       011     6    FORMAT (/28H NO. OF NEG DATA DISCARDED =,I4/)
000017       011     7    FORMAT (/26H GENERAL STATISTICS:  N = I4,4X6HRMSE =E10.3,4X4H R =
000018       011         1E10.3//25X4HSO =,E10.3,4X,4HSP =,E10.3//
000019       011         232H 95 PERCENT CONFIDENCE INTERVAL:E10.3,12H .LE.RHO.LE.,E10.3//
000020       011         317H REGRESSION LINE:E10.3,3H + E10.3,4H * X,10X,4HS = E10.3/)
000021       011     8    FORMAT (//32H SENSITIVITY CASE STATISTICS .../)
000022       012    9     FORMAT(I2,1X,10A6)
000023       011    C
000024       011    C     INPUT FROM FARAMETER FILE ...
000025       011    C
000026       011    C           IDEN  = DATA IDENTIFIER.
000027       011    C           ICODE = 0, NO GRAPHICAL OUTPUT.
000028       011    C                 = 1, PLOT SCATTER GRAPH.
000029       011    C                 = 2, ADD THE REGRESSION LINE TO CODE 1.
000030       011    C                 = 3, ADD THE SENSITIVITY BOUNDS TO CODES 1-2.
000031       011    C                 = 4, ADD THE CONFIDENCE BOUNDS TO CODES 1-2.
000032       011    C                 = 5, ADD THE PROBABILITY BOUNDS TO CODES 1-2.
000033       011    C                 = 6, ADD THE CONFIDENCE AND PROBABILITY BOUNDS TO CODES 1-2.
000034       011    C           ITYPE = 0001, X-AXIS IS LINEAR.
000035       011    C                 = 0010, Y-AXIS IS LINEAR
000036       011    C                 = 0100, X-AXIS IS LOGARITHMIC.
000037       011    C                 = 1000, Y-AXIS IS LOGARITHMIC.
000038       011    C           NTC   = NO. OF ASCII CHARACTERS IN THE TITLE.
000039       011    C           NXC   = NO. OF ASCII CHARACTERS IN THE X-AXIS LABEL.
000040       011    C           NYC   = NO. OF ASCII CHARACTERS IN THE Y-AXIS LABEL.
000041       011    C           ZARR  = GRAPH LOWER TITLE.
000042       011    C           XARR  = X-AXIS LABEL.
000043       011    C           YARR  = Y-AXIS LABEL.
000044       011    C           TAB   = STUDENT-T TABLE FOR 90 PERCENT C.L. AND N-2 D.O.F.
000045       011    C
000046       011          READ 2,IDEN
000047       011    CR    PRINT 1,IDEN
000048       011          READ 3,ICODE,ITYPE
000049       011          READ 3,(TAB(L),L=1,34)
000050       011          READ 9,NTC,TIT
000051       011          READ 9,NXC,XARR
000052       011          READ 9,NYC,YARR
000053       011    CR    PRINT5,ICODE,ITYPF,NTC,NXC,NYC
000054       011          DO 60 I=1,6
000055       011          IF (I.NE.ICODE) GO TO 50
```

```
011           READ 9,NZC,ZARR
011           GO TO 60
011      50   READ 9,NZ,ZAR
011      60   CONTINUE
011     C
011     C     INPUT FROM DATA BASE FILE ...
011     C
011     C           ND = NO. OF DATA.
011     C           NS = NO. OF STATIONS.
011     C           IDATE = DATE OF OBSERVATION
011     C           ISITE = STATION NO.
011     C           ITIME = TIME OF OBSERVATION
011     C           ISTC,IWSC,IWDC ARE NOT USED.
011     C           OC = OBSERVED CONCENTRATION
011     C           PC = PREDICTED CONCENTRATION
011     C
011     100   READ 3,ND,NS
011     CR    PRINT 4,ND,NS
011           N=0
011           DO 120 I=1,ND
011           READ 3,IDATE(I),ITIME(I),ISITE(I),ISTC,IWSC,IWDC,OC(I),PC(I)
011           IF (OC(I).LT.0.0.OR.PC(I).LT.0.0) GO TO 115
011           N=N+1
011           OC(N)=OC(I)
011           PC(N)=PC(I)
011           YMAX=AMAX1(YMAX,OC(N),PC(N))
011           IF (ISITE(I).GT.1) GO TO 105
011           N1=N1+1
011           X1(N1)=OC(I)
011           Y1(N1)=PC(I)
011           IF (Y1(N1).GT.YBIG) YBIG=Y1(N1)
011           GO TO 120
011     105   IF (ISITE(I).GT.2) GO TO 110
011           N2=N2+1
011           X2(N2)=OC(I)
011           Y2(N2)=PC(I)
011           IF (Y2(N2).GT.YBIG) YBIG=Y2(N2)
011           GO TO 120
011     110   N3=N3+1
011           X3(N3)= OC(I)
011           Y3(N3)=PC(I)
011           IF (Y3(N3).GT.YBIG) YBIG=Y3(N3)
011           GO TO 120
011     115   NEG=NEG+1
011     CR    PRINT 6,NEG
011     120   CONTINUE
011           XMAX=YMAX
011           CALL LINLSQ (OC,PC,N,YAL)
011     CR    PRINT 7,N,RMSE,RC,XSD,YSD,CIL,CIU,A,B,ESD
011           NN=100
011           DELX=(XMAX-XMIN)/NN
011           XX(1)=XMIN
011           YY(1)=A+B*XX(1)
011           DO 125 I=2,NN
011           XX(I)=XX(I-1)+DELX
011           YY(I)=A+B*XX(I)
011     125   CONTINUE
```

```
011             IF (ICODE.EQ.0) GO TO 130
011             IF (ICODE.NE.3) GO TO 145
011   C
011   C              PLOT THE SENSITIVITY BOUNDS (0.1*YMAX).
011   C
011    130      CONTINUE
011   CR        PRINT 8
011             YFACT=0.1*YPIG
011             DO 135 I=1,NN
011             YPLP(I)=YY(I)+YFACT
011             YPLM(I)=YY(I)-YFACT
011     135     CONTINUE
011             M=0
011             DO 140 I=1,N
011             YP=YRL(I)+YFACT
011             YM=YRL(I)-YFACT
011             IF (PC(I).LT.YP.AND.PC(I).GT.YM) GO TO 140
011             M=M+1
011             XS(M )=OC(I)
011             YS(M )=PC(I)
011     140     CONTINUE
011             CALL LINLSQ (XS,YS,M ,YRL)
011   CR        PRINT 7,M,RMSE,RC,XSD,YSD,CIL,CIU,A,B,ESD
011             IF (ICODE.EQ.0) STOP
011   C
011   C              PLOT THE SCATTER GRAPH.
011   C
011   C              INITIALIZE TEKTRONIX GRAPH PACKAGE.
011     145     CALL INITT (120)
011             CALL TERM(2,1024)
011             CALL HEADER
011             CALL CHRSIZ(2)
011             CALL BOX(0,0,1023,780)
011             CALL MOVABS (10,760)
011             CALL AOUTST(33,'LINEAR REGRESSION ANALYSIS, CASE ')
011             ENCODE(1,99,CASE) ICODE
011    99       FORMAT(I1)
011             CALL AOUTST(1,CASE)
011   C PRINT LEGENDS
011             CALL CHRSIZ(3)
011             AN=N
011             IF(ICODE.EQ.3) AN=M
011             ENCODE(48,95,AOUT) B,A,RC,AN
012    95       FORMAT(4(1PG8.2,4X))
011             CALL MOVABS(4,4)
011             CALL AOUTST(3,'N =')
011             CALL AOUTST(8,AOUT(7))
011             CALL MOVABS(4,18)
011             CALL AOUTST(3,'R =')
011             CALL AOUTST(8,AOUT(5))
011             CALL MOVABS(4,32)
011             CALL AOUTST(3,'B =')
011             CALL AOUTST(8,AOUT(3))
011             CALL MOVABS(4,46)
011             CALL AOUTST(3,'M =')
011             CALL AOUTST(8,AOUT(1))
011             CALL BOX(0,0,100,60)
```

```
000170      011          IF(ICODE.EQ.1.OR.ICODE.EQ.5) GO TO 80
000171      011          CALL MOVABS(720,4)
000172      011          CALL AOUTST(31,'SOLID LINE - LEAST SQUARES LINE')
000173      011          CALL MOVABS(720,18)
000174      011          IF(ICODE.EQ.4) CALL AOUTST(33,'DASH LINE - 90% CONFIDENCE BOUNDS')
000175      011          IF(ICODE.EQ.2) CALL AOUTST(22,'DASH LINE - IDEAL LINE')
000176      011          IF(ICODE.EQ.3) CALL AOUTST(30,'DASH LINE - SENSITIVITY BOUNDS')
000177      014          IF(ICODE.NE.6) CALL BOX(716,0,1023,32)
000178      014          IF(ICODE.EQ.3) GO TO 85
000179      011          IF(ICODE.NE.6) GO TO 80
000180      011          CALL AOUTST(34,'SMALL DASH - 90% CONFIDENCE BOUNDS')
000181      011          CALL MOVABS(720,32)
000182      011          CALL AOUTST(35,'LARGE DASH - 90% PROBABILITY BOUNDS')
000183      011          CALL BOX(716,0,1023,46)
000184      014          GO TO 80
000185      015    85    CALL MOVABS(104,4)
000186      011          CALL AOUTST(63,'NOTE: PARAMETERS M,B,R,N ARE FOR POINTS OUTSIDE DA
000187      011         1SHED BOUNDARY')
000188      013    80    CALL BINITT
000189      011          CALL CHRSIZ(2)
000190      011    C           SET THE SCREEN LIMITS FOR X AND Y.
000191      011          CALL PLACE (3HSTD)
000192      011          CALL XFRM (3)
000193      011          CALL YFRM (3)
000194      011          CALL XMFRM (3)
000195      011          CALL YMFRM (3)
000196      011          CALL MOVABS(300,50)
000197      011          CALL AOUTST(NXC,XARR)
000198      011          CALL QUOTE(50.,200.,.15,YARR,90.,NYC)
000199      011          CALL MOVABS(300,730)
000200      011          CALL AOUTST(NTC,TIT)
000201      011    C           DRAW A STANDARD FRAME.
000202      011          CALL FRAME
000203      011    C           PLOT THE MAJOR-MINOR TIC MARKS ON THE AXES.
000204      011          CALL DLIMX (XMIN,XMAX)
000205      011          CALL DLIMY (YMIN,YMAX)
000206      011          CALL XNEAT (1)
000207      011          CALL YNEAT (1)
000208      011    C           DETERMINE THE (X,Y) AXES TYPE.
000209      011          IF (ITYPE.GT.100) CALL YTYPE(2)
000210      011          ITYPE=ITYPE-1000
000211      011          IF (ITYPE.GT.10) CALL XTYPE(2)
000212      011          ITYPE=ITYPE-100
000213      011          IF (ITYPE.GT.1) CALL YTYPE(1)
000214      011          ITYPE=ITYPE-10
000215      011          IF (ITYPE.GT.0) CALL XTYPE(1)
000216      011    C           1ST STATION (TPIANGLES).
000217      011          CALL STEPS (1)
000218      011          CALL LINE (-1)
000219      011          CALL SIZES (0.5)
000220      011          CALL SYMBL (3)
000221      011          CALL NPTS (N1)
000222      011          CALL CHECK (X1,Y1)
000223      011          CALL DSPLAY (X1,Y1)
000224      011    C           2ND STATION (DIAMONDS).
000225      011          CALL SYMBL (6)
000226      011          CALL NPTS (N2)
```

```
011                CALL CPLOT (X2,Y2)
011   C                  3RD STATION (SQUARES).
011                CALL SYMBL (4)
011                CALL NPTS (N3)
011                CALL CPLOT (X3,Y3)
011                IF (ICODE.EQ.1) GO TO 180
011   C
011   C                  PLOT THE REGRESSION LINE (SOLID).
011   C
011                CALL STEPL (1)
011   CR ORIGINAL AGII MANUAL USED 0 FOR SOLID LINE
011                CALL LINE (0)
011                CALL NPTS (NN)
011                CALL SYMBL (0)
011                CALL CPLOT (XX,YY)
011                IF (ICODE.GT.2) GO TO 150
011   C
011   C                  PLOT THE PERFECT LINE (DASH).
011   C
011                XX(1)=XMIN
011                YY(1)=XMIN
011                XX(2)=XMAX
011                YY(2)=XMAX
011                CALL LINE (12)
011                CALL NPTS (2)
011                CALL CPLOT (XX,YY)
011                GO TO 180
011           150  IF (ICODE.EQ.3) GO TO 175
011                RECNP1=RECNOP+1
011                FACT1=STAB(N,TAB)*ESD
011                IF (ICODE.EQ.5) GO TO 165
011   C
011   C                  PLOT THE 90 PERCENT CONFIDENCE BOUNDS.
011   C
011                DO 160 I=1,NN
011                XMXB=XX(I)-XBAR
011                DIVI=XMXB*XMXB/SXMXP
011                FACT2=SQRT(RECNOP+DIVI)
011                FACT3=FACT1*FACT2
011                YCLP(I)=YY(I)+FACT3
011                YCLM(I)=YY(I)-FACT3
011           160  CONTINUE
011                CALL LINE(12)
011                CALL NPTS(NN)
011                CALL CPLOT (XX,YCLP)
011                CALL CPLOT (XX,YCLM)
011                IF (ICODE.EQ.4) GO TO 180
011   C
011   C                  PLOT THE PROBABILITY BOUNDS.
011   C
011           165  DO 170 I=1,NN
011                XMXB=XX(I)-XBAR
011                DIVI=XMXB*XMXB/SXMXP
011                FACT6=SQRT(RECNP1+DIVI)
011                FACT7=FACT1*FACT6
011                YPLP(I)=YY(I)+FACT7
011                YPLM(I)=YY(I)-FACT7
```

```
000284       011     170  CONTINUE
000285       011     175  CALL LINE (34)
000286       011          CALL NPTS (NN)
000287       011          CALL CPLOT (XX,YPLP)
000288       011          CALL CPLOT (XX,YPLM)
000289       011     180  CONTINUE
000290       011          CALL HDCOPY
000291       011          CALL FINITT (0,700)
000292       011     C         END OF PLOT.
000293       011          END

END ELT.
```

F.4 Subroutine M21ADO*STAT04.LINLSQ

```
@ELT,L S4.LINLSQ
ELT007 SL73R1 11/01/78 09:43:56 (0,)
000001      000          SUBROUTINE LINLSQ (X,Y,NOD,YRL)
000002      000    C
000003      000    C          YBAR = A + B*XBAR
000004      000    C
000005      000          COMMON /LSQ/ RECNOP,A,B,XSD,YSD,CIL,CIU,SXMXB,RC,RMSE,ESD,XBAR
000006      000          DIMENSION X(NOD),Y(NOD),YRL(NOD)
000007      000          SUMX=0.0
000008      000          SUMY=0.0
000009      000          SUMXY=0.0
000010      000          SUMX2=0.0
000011      000          SUMY2=0.0
000012      000          SMXMY2=0.0
000013      000          NBAD=0
000014      000          DO 100 I=1,NOD
000015      000          IF (X(I).LT.0.0.OR.Y(I).LT.0.0) GO TO 50
000016      000          SUMX=SUMX+X(I)
000017      000          SUMY=SUMY+Y(I)
000018      000          SUMXY=SUMXY+X(I)*Y(I)
000019      000          SUMX2=SUMX2+X(I)*X(I)
000020      000          SUMY2=SUMY2+Y(I)*Y(I)
000021      000          SMXMY2=SMXMY2+(X(I)-Y(I))*(X(I)-Y(I))
000022      000          GO TO 100
000023      000    50    NBAD=NBAD+1
000024      000    100   CONTINUE
000025      000          NOP=NOD-NBAD
000026      000          RECNOP=1.0/NOP
000027      000          XBAR=SUMX*RECNOP
000028      000          YBAR=SUMY*RECNOP
000029      000          ARG1=SUMX2*RECNOP-XBAR*XBAR
000030      000          XSD=SQRT(ARG1)
000031      000          ARG2=SUMY2*RECNOP-YBAR*YBAR
000032      000          YSD=SQRT(ARG2)
000033      000          B=(NOP*SUMXY-SUMX*SUMY)/(SUMX2*NOP-SUMX*SUMX)
000034      000          A=YBAR-B*XBAR
000035      000          RC=(SUMXY*RECNOP-XBAR*YBAR)/SQRT(ARG1*ARG2)
000036      000          RMSE=SQRT(SMXMY2*RECNOP)
000037      000          SUM2=0.0
000038      000          SXMXB=0.0
000039      000          DO 105 J=1,NOP
000040      000          YRL(J)=A+B*X(J)
000041      000          YMYRL=Y(J)-YRL(J)
000042      000          SUM2=SUM2+YMYRL*YMYRL
000043      000          XMXB=X(J)-XBAR
000044      000          XMXB2=XMXB*XMXB
000045      000          SXMXB=SXMXB+XMXB2
000046      000    105   CONTINUE
000047      000          EVAR=SUM2/(NOP-2)
000048      000          ESD=SQRT(EVAR)
000049      000          CON=1.96/SQRT(NOP-3.0)
000050      000          AHBRC=0.5*ALOG((1.0+RC)/(1.0-RC))
000051      000          CIL=TANH(AHBRC-CON)
000052      000          CIU=TANH(AHBRC+CON)
000053      000          RETURN
000054      000          END
```

F.4 Subroutine M21ADO*STAT04.STAB

```
@ELT,L S4.STAB
ELT007 SL73R1 11/01/78 09:44:14 (0,)
          000          FUNCTION STAB (IND,TAB)
          000    C
          000    C           ESTIMATION OF POPULATION MEAN USING THE STUDENT-T DISTRIBUTION
          000    C     WITH 90 PERCENT CONFIDENCE INTERVAL AND 2 DEGREES OF FREEDOM.
          000    C
          000          DIMENSION TAB(34)
          000          NU=IND-2
          000          IF (NU.GT.0) GO TO 100
          000          PRINT 1,NU
          000    1     FORMAT (/47H NU.LT.1 STUDENT-T VALUE CAN NOT BE DETERMINED. )
          000          STOP
          000    100   IF (NU.GT.30) GO TO 110
          000          STAB=TAB(NU)
          000          RETURN
          000    110   IF (NU.GT.40) GO TO 120
          000          STAB=TAB(31)
          000          RETURN
          000    120   IF (NU.GT.60) GO TO 130
          000          STAB=TAB(32)
          000          RETURN
          000    130   IF (NU.GT.120) GO TO 140
          000          STAB=TAB(33)
          000          RETURN
          000    140   STAB=TAB(34)
          000          RETURN
          000          END

END ELT.
```

APPENDIX G

INTERSTATION ERROR CORRELATION LISTINGS

G.1 Program Map, M21ADO*STAT05.MAPIT

```
@ELT,L S5.MAPIT
ELT007 SL73R1 11/01/78 09:46:01 (0,)
000001        000   IN STAT05.COREL2
000002        000   IN STAT05.LINLSQ
000003        000   IN STAT05.XYCORR
000004        000   END

END ELT.
```

G.2 Listing of the Main Program, M21ADO*STAT05.COREL2

```
@ELT,L S5.COREL2
ELT007 SL73R1 11/01/78 09:45:50 (1,)
000001       000           COMMON /LSQ/ RECNOP,A,B,XSD,YSD,CIL,CIU,SXRXY,NC,RMSE,ESD
000002       001           DIMENSION PC(123,3),OC(123,3),IDATE(123),ITIME(123),CONR(3,3),CONL
000003       001          1(3,3),CONU(3,3)
000004       001      1    FORMAT (//35H INTER-STATION ERROR CORRELATION TESTS./)
000005       001      2    FORMAT (/13H NO. OF DATA:I5,5X,13HNO. OF SITES:,I5,5X,5HIOPT:,I3,
000006       001          15X,6HSITE1:I5,5X6HSITE2:,I5/)
000007       001      3    FORMAT ()
000008       001      4    FORMAT (/16H NO. OF GOOD DATA:,I5,5X12HCORRELATION:,F5.2,5X,18HCON
000009       000          1FIDENCE LIMITS:,F6.2,12H .LE.RHO.LE.,F6.2,/)
000010       001      5    FORMAT (//23H CORRELATION MATRIX ...,//5H SITE,25I5)
000011       001      6    FORMAT (/1H ,I3,2X,25F5.2)
000012       001      7    FORMAT (//40H 95-PERCENT CONFIDENCE LIMITS MATRIX ...,//5H SITE,
000013       001          125I5)
000014       001      8    FORMAT (4X1HU,1X,25F5.2)
000015       001      9    FORMAT (/1H ,I2,1X,1HL,1X,25F5.2)
000016       000     C
000017       000     C     INPUT FROM DATA BASE FILE ...
000018       000     C
000019       001     C           ND   = NO. OF DATA
000020       001     C           NS   = NO. OF STATIONS
000021       001     C           IOPT = 1, CORRELATION BETWEEN 2 STATIONS ONLY.
000022       001     C                = 2,  CORRELATION AND CONFIDENCE LIMITS MATRICES.
000023       001     C           IS1  = STATION NO. 1 WHEN IOPT=1.
000024       001     C           IS2  = STATION NO. 2 WHEN IOPT=1.
000025       001     C           IDATES = DATE OF OBSERVATION
000026       001     C           ITIMES = TIME OF OBSERVATION
000027       001     C           ISITE = SITE OF OBSERVATION
000028       000     C           ISTC, IWSC, IWDC NOT USED.
000029       001     C           OCS   = OBSERVED CONCENTRATION
000030       001     C           PCS   = PREDICTED CONCENTRATION
000031       000     C
000032       000           PRINT 1
000033       000           READ 3,ND,NS,IOPT,IS1,IS2
000034       000           PRINT 2,ND,NS,IOPT,IS1,IS2
000035       001           DO 110 I=1,ND
000036       001           DO 100 J=1,NS
000037       001           OC(I,J)=-99.9
000038       001           PC(I,J)=-99.9
000039       000    100    CONTINUE
000040       000    110    CONTINUE
000041       000     C
000042       000     C           RETAIN OC AND PC BY SITE NO. AND TIME.
000043       000     C
000044       001           NT=1
000045       001           DO 135 I=1,ND
000046       000           READ 3,IDATES,ITIMES,ISITE,ISTC,IWSC,IWDC,OCS,PCS
000047       001           IF (I.GT.1) GO TO 115
000048       001           IDATED=IDATES
000049       001           ITIMED=ITIMES
000050       001    115    IF (IDATES.EQ.IDATED) GO TO 120
000051       001           IDATED=IDATES
000052       001           NT=NT+1
000053       001           IDATE (NT)=IDATED
000054       000           GO TO 125
000055       001    120    IF (ITIMES.EQ.ITIMED) GO TO 130
```

```
001            NT=NT+1
001      125   ITIMED=ITIMES
001            ITIME (NT)=ITIMED
001      130   IF (OCS.LT.0.0.OR.PCS.LT.0.0) GO TO 135
001            PC(NT,ISITE)=PCS
001            OC(NT,ISITE)=OCS
000      135   CONTINUE
001            IF (IOPT.GT.1) GO TO 200
000     C
000     C            OPTION 1: DETERMINE THE CORRELATION BETWEEN ANY TWO SITES.
000     C
001            CALL XYCORR (NT,OC(1,IS1),PC(1,IS1),OC(1,IS2),PC(1,IS2))
000            PRINT 4,NT,RC,CIL,CIU
000            STOP1
000     C
000     C            OPTION 2: DETERMINE THE CORRELATION AND CONFIDENCE LIMITS MATRIX.
000     C
001      200   DO 230 I=1,NS
001            DO 220 J=1,NS
001            IF (I.EQ.J) GO TO 210
001            CALL XYCORR(NT,OC(1,I),PC(1,I),OC(1,J),PC(1,J))
001            CORM(I,J)=RC
001            CONL(I,J)=CIL
001            CONU(I,J)=CIU
000            GO TO 220
001      210   CORM(I,J)=1.0
001            CONL(I,J)=1.0
001            CONU(I,J)=1.0
000      220   CONTINUE
000      230   CONTINUE
000     C
000     C            PRINT OUT THE CORRELATION MATRIX.
000     C
001            PRINT 5,(I,I=1,NS)
001            DO 240 I=1,NS
001            PRINT 6,I,(CORM(I,J),J=1,NS)
000      240   CONTINUE
000     C
000     C            PRINT OUT THE LOWER AND UPPER CONFIDENCE LIMITS.
000     C
001            PRINT 7,(I,I=1,NS)
001            DO 250 I=1,NS
001            PRINT 9,I,(CONL(I,J),J=1,NS)
001            PRINT 8,(CONU(I,J),J=1,NS)
000      250   CONTINUE
000            STOP2
000            END

END ELT.
```

G.3 Subroutine M21ADO*STAT05.XYCORR

```
@ELT,L S5.XYCORR
ELTC07 SL73R1 11/01/78 09:46:03 (1,)
       001            SUBROUTINE XYCORR (NT,OC1,PC1,OC2,PC2)
       000            COMMON /LSQ/ RECNOP,A,B,XSD,YSD,CIL,CIU,SXMX6,RC,RMSE,ESD
       001            DIMENSION OC1(NT),PC1(NT),OC2(NT),PC2(NT),DC1(123),DC2(123),YRL(12
       001           13)
       000     C
       001     C            COMPUTE (OC-PC) FOR FIRST SITE.
       000     C
       001            NN=0
       001            DO 100 J=1,NT
       001            IF (OC1(J).LT.0.0.OR.PC1(J).LT.0.0.OR.OC2(J).LT.0.0.OR.PC2(J).LT.
       001           10.0) GO TO 100
       001            NN=NN+1
       001            DC1(NN)=OC1(J)-PC1(J)
       001            DC2(NN)=OC2(J)-PC2(J)
       000       100  CONTINUE
       000     C
       000     C            COMPUTE THE LINEAR CORRELATION BETWEEN TWO SITES.
       000     C
       001            CALL LINLSQ (DC1,DC2,NN,YRL)
       000            RETURN
       000            END

END ELT.
```